

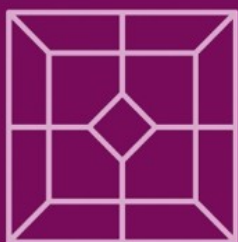
Tomáš Pajdla
Jiří Matas (Eds.)

LNC3 3021

Computer Vision – ECCV 2004

8th European Conference on Computer Vision
Prague, Czech Republic, May 2004
Proceedings, Part I

1 Part I



ECCV 2004



Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

University of Dortmund, Germany

Madhu Sudan

Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos

New York University, NY, USA

Doug Tygar

University of California, Berkeley, CA, USA

Moshe Y. Vardi

Rice University, Houston, TX, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

Springer

Berlin

Heidelberg

New York

Hong Kong

London

Milan

Paris

Tokyo

Tomáš Pajdla Jiří Matas (Eds.)

Computer Vision – ECCV 2004

8th European Conference on Computer Vision
Prague, Czech Republic, May 11-14, 2004
Proceedings, Part I



Springer

Volume Editors

Tomáš Pajdla

Jiří Matas

Czech Technical University in Prague, Department of Cybernetics

Center for Machine Perception

121-35 Prague 2, Czech Republic

E-mail: {pajdla,matas}@cmp.felk.cvut.cz

Library of Congress Control Number: 2004104846

CR Subject Classification (1998): I.4, I.3.5, I.5, I.2.9-10

ISSN 0302-9743

ISBN 3-540-21984-6 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable to prosecution under the German Copyright Law.

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2004

Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 11007678 06/3142 5 4 3 2 1 0

Preface

Welcome to the proceedings of the 8th European Conference on Computer Vision!

Following a very successful ECCV 2002, the response to our call for papers was almost equally strong – 555 papers were submitted. We accepted 41 papers for oral and 149 papers for poster presentation.

Several innovations were introduced into the review process. First, the number of program committee members was increased to reduce their review load. We managed to assign to program committee members no more than 12 papers. Second, we adopted a paper ranking system. Program committee members were asked to rank all the papers assigned to them, even those that were reviewed by additional reviewers. Third, we allowed authors to respond to the reviews consolidated in a discussion involving the area chair and the reviewers. Fourth, the reports, the reviews, and the responses were made available to the authors as well as to the program committee members. Our aim was to provide the authors with maximal feedback and to let the program committee members know how authors reacted to their reviews and how their reviews were or were not reflected in the final decision. Finally, we reduced the length of reviewed papers from 15 to 12 pages.

The preparation of ECCV 2004 went smoothly thanks to the efforts of the organizing committee, the area chairs, the program committee, and the reviewers. We are indebted to Anders Heyden, Mads Nielsen, and Henrik J. Nielsen for passing on ECCV traditions and to Dominique Asselineau from ENST/TSI who kindly provided his GestRFIA conference software. We thank Jan-Olof Eklundh and Andrew Zisserman for encouraging us to organize ECCV 2004 in Prague. Andrew Zisserman also contributed many useful ideas concerning the organization of the review process. Olivier Faugeras represented the ECCV Board and helped us with the selection of conference topics. Kyros Kutulakos provided helpful information about the CVPR 2003 organization. David Vernon helped to secure ECVision support.

This conference would never have happened without the support of the Centre for Machine Perception of the Czech Technical University in Prague. We would like to thank Radim Šára for his help with the review process and the proceedings organization. We thank Daniel Večerka and Martin Matoušek who made numerous improvements to the conference software. Petr Pohl helped to put the proceedings together. Martina Budošová helped with administrative tasks. Hynek Bakstein, Ondřej Chum, Jana Kostková, Branislav Mičušík, Štěpán Obdržálek, Jan Šochman, and Vít Zýka helped with the organization.

Organization

Conference Chair

Václav Hlaváč

CTU Prague, Czech Republic

Program Chairs

Tomáš Pajdla

CTU Prague, Czech Republic

Jiří Matas

CTU Prague, Czech Republic

Organization Committee

Tomáš Pajdla

CTU Prague, Czech Republic

Radim Šára

Workshops, Tutorials

CTU Prague, Czech Republic

Vladimír Smutný

Budget, Exhibition

CTU Prague, Czech Republic

Eva Matysková

Local Arrangements

CTU Prague, Czech Republic

Jiří Matas

CTU Prague, Czech Republic

Václav Hlaváč

CTU Prague, Czech Republic

Conference Board

Hans Burkhardt

University of Freiburg, Germany

Bernard Buxton

University College London, UK

Roberto Cipolla

University of Cambridge, UK

Jan-Olof Eklundh

Royal Institute of Technology, Sweden

Olivier Faugeras

INRIA, Sophia Antipolis, France

Anders Heyden

Lund University, Sweden

Bernd Neumann

University of Hamburg, Germany

Mads Nielsen

IT University of Copenhagen, Denmark

Giulio Sandini

University of Genoa, Italy

David Vernon

Trinity College, Ireland

Area Chairs

Dmitry Chetverikov

MTA SZTAKI, Hungary

Kostas Daniilidis

University of Pennsylvania, USA

Rachid Deriche

INRIA Sophia Antipolis, France

Jan-Olof Eklundh

KTH Stockholm, Sweden

Luc Van Gool

KU Leuven, Belgium & ETH Zürich, Switzerland

Richard Hartley

Australian National University, Australia

Michal Irani	Weizmann Institute of Science, Israel
Sing Bing Kang	Microsoft Research, USA
Aleš Leonardis	University of Ljubljana, Slovenia
Stan Li	Microsoft Research China, Beijing, China
David Lowe	University of British Columbia, Canada
Mads Nielsen	IT University of Copenhagen, Denmark
Long Quan	HKUST, Hong Kong, China
Jose Santos-Victor	Instituto Superior Tecnico, Portugal
Cordelia Schmid	INRIA Rhône-Alpes, France
Steven Seitz	University of Washington, USA
Amnon Shashua	Hebrew University of Jerusalem, Israel
Stefano Soatto	UCLA, Los Angeles, USA
Joachim Weickert	Saarland University, Germany
Andrew Zisserman	University of Oxford, UK

Program Committee

Jorgen Ahlberg	Joachim Buhmann	Alexei Efros
Narendra Ahuja	Hans Burkhardt	Irfan Essa
Yiannis Aloimonos	Aurelio Campilho	Michael Felsberg
Arnon Amir	Octavia Camps	Cornelia Fermueller
Elli Angelopoulou	Stefan Carlsson	Mario Figueiredo
Helder Araujo	Yaron Caspi	Bob Fisher
Tal Arbel	Tat-Jen Cham	Andrew Fitzgibbon
Karl Astrom	Mike Chantler	David Fleet
Shai Avidan	Francois Chaumette	Wolfgang Foerstner
Simon Baker	Santanu Choudhury	David Forsyth
Subhashis Banerjee	Laurent Cohen	Pascal Fua
Kobus Barnard	Michael Cohen	Dariu Gavrilă
Ronen Basri	Bob Collins	Jan-Mark Geusebroek
Serge Belongie	Dorin Comaniciu	Christopher Geyer
Marie-Odile Berger	Tim Cootes	Georgy Gimelfarb
Horst Bischof	Joao Costeira	Frederic Guichard
Michael J. Black	Daniel Cremers	Gregory Hager
Andrew Blake	Antonio Criminisi	Allan Hanbury
Laure Blanc-Feraud	James Crowley	Edwin Hancock
Aaron Bobick	Kristin Dana	Horst Haussecker
Rein van den Boomgaard	Trevor Darrell	Eric Hayman
Terrance Boulton	Larry Davis	Martial Hebert
Richard Bowden	Fernando De la Torre	Bernd Heisele
Edmond Boyer	Frank Dellaert	Anders Heyden
Mike Brooks	Joachim Denzler	Adrian Hilton
Michael Brown	Greg Dudek	David Hogg
Alfred Bruckstein	Chuck Dyer	Atsushi Imiya

Michael Isard	Nassir Navab	Jon Sporning
Yuri Ivanov	Shree Nayar	Charles Stewart
David Jacobs	Ko Nishino	Peter Sturm
Allan D. Jepson	David Nister	Changming Sun
Peter Johansen	Ole Fogh Olsen	Tomas Svoboda
Nebojsa Jojic	Theodore Papadopoulos	Rahul Swaminathan
Frederic Jurie	Nikos Paragios	Richard Szeliski
Fredrik Kahl	Shmuel Peleg	Tamas Sziranyi
Daniel Keren	Francisco Perales	Chi-keung Tang
Benjamin Kimia	Nicolas Perez	Hai Tao
Ron Kimmel	de la Blanca	Sibel Tari
Nahum Kiryati	Pietro Perona	Chris Taylor
Georges Koepfler	Matti Pietikainen	C.J. Taylor
Pierre Kornprobst	Filiberto Pla	Bart ter Haar Romeny
David Kriegman	Robert Pless	Phil Torr
Walter Kropatsch	Marc Pollefeys	Antonio Torralba
Rakesh Kumar	Jean Ponce	Panos Trahanias
David Liebowitz	Ravi Ramamoorthi	Bill Triggs
Tony Lindeberg	James Rehg	Emanuele Trucco
Jim Little	Ian Reid	Dimitris Tsakiris
Yanxi Liu	Tammy Riklin-Raviv	Yanghai Tsin
Yi Ma	Ehud Rivlin	Matthew Turk
Claus Madsen	Nicolas Rougon	Tinne Tuytelaars
Tom Malzbender	Yong Rui	Nuno Vasconcelos
Jorge Marques	Javier Sanchez	Baba C. Vemuri
David Marshall	Guillermo Sapiro	David Vernon
Bogdan Matei	Yoichi Sato	Alessandro Verri
Steve Maybank	Eric Saund	Rene Vidal
Gerard Medioni	Otmar Scherzer	Jordi Vitria
Etienne Memin	Bernt Schiele	Yair Weiss
Rudolf Mester	Mikhail Schlesinger	Tomas Werner
Krystian Mikolajczyk	Christoph Schnoerr	Carl-Fredrik Westin
J.M.M. Montiel	Stan Sclaroff	Ross Whitaker
Theo Moons	Mubarak Shah	Lior Wolf
Pavel Mrazek	Eitan Sharon	Ying Wu
Joe Mundy	Jianbo Shi	Ming Xie
Vittorio Murino	Kaleem Siddiqi	Ramin Zabih
David Murray	Cristian Sminchisescu	Assaf Zomet
Hans-Hellmut Nagel	Nir Sochen	Steven Zucker
Vic Nalwa	Gerald Sommer	
P.J. Narayanan	Gunnar Sparr	

Additional Reviewers

Lourdes Agapito	Claudio Fanti	Jocelyn Marchadier
Manoj Aggarwal	Michela Farenzena	Scott McCloskey
Parvez Ahammad	Doron Feldman	Leonard McMillan
Fernando Alegre	Darya Frolova	Marci Meingast
Jonathan Alon	Andrea Fusiello	Anurag Mittal
Hans Jorgen Andersen	Chunyu Gao	Thomas B. Moeslund
Marco Andreetto	Kshitiz Garg	Jose Montiel
Anelia Angelova	Yoram Gat	Philippos Mordohai
Himanshu Arora	Dan Gelb	Pierre Moreels
Thangali Ashwin	Ya'ara Goldschmidt	Hesam Najafi
Vassilis Athitsos	Michael E. Goss	P.J. Narayanan
Henry Baird	Leo Grady	Ara Nefian
Harlyn Baker	Sertan Grigin	Oscar Nestares
Evgeniy Bart	Michael Grossberg	Michael Nielsen
Moshe Ben-Ezra	J.J. Guerrero	Peter Nillius
Manuele Bicego	Guodong Guo	Fredrik Nyberg
Marten Björkman	Yanlin Guo	Tom O'Donnell
Paul Blaer	Robert Hanek	Eyal Ofek
Ilya Blayvas	Matthew Harrison	Takahiro Okabe
Eran Borenstein	Tal Hassner	Kazunori Okada
Lars Bretzner	Horst Haussecker	D. Ortin
Alexia Briassouli	Yakov Hel-Or	Patrick Perez
Michael Bronstein	Anton van den Hengel	Christian Perwass
Rupert Brooks	Tat Jen Cham	Carlos Phillips
Gabriel Brostow	Peng Chang	Srikumar Ramalingam
Thomas Brox	John Isidoro	Alex Rav-Acha
Stephanie Brubaker	Vishal Jain	Stefan Roth
Andres Bruhn	Marie-Pierre Jolly	Ueli Rutishauser
Darius Burschka	Michael Kaess	C. Sagues
Umberto Castellani	Zia Khan	Garbis Salgian
J.A. Castellanos	Kristian Kirk	Ramin Samadani
James Clark	Dan Kong	Bernard Sarel
Andrea Colombari	B. Kröse	Frederik Schaffalitzky
Marco Cristani	Vivek Kwatra	Adam Seeger
Xiangtian Dai	Michael Langer	Cheng Dong Seon
David Demirdjian	Catherine Laporte	Ying Shan
Maxime Descoteaux	Scott Larsen	Eli Shechtman
Nick Diakopoulous	Barbara Levienaise-	Grant Schindler
Anthony Dicks	Obadia	Nils T. Siebel
Carlotta Domeniconi	Frederic Leymarie	Leonid Sigal
Roman Dvrgard	Fei-Fei Li	Greg Slabaugh
R. Dugad	Rui Li	Ben Southall
Ramani Duraiswami	Kok-Lim Low	Eric Spellman
Kerrien Erwan	Le Lu	Narasimhan Srinivasa

Drew Steedly	Zhizhou Wang	Ruigang Yang
Moritz Stoerring	Joost van de Weijer	Yll Haxhimusa
David Suter	Wolfgang Wein	Tianli Yu
Yi Tan	Martin Welk	Lihi Zelnik-Manor
Donald Tanguay	Michael Werman	Tao Zhao
Matthew Toews	Horst Wildenauer	Wenyi Zhao
V. Javier Traver	Christopher R. Wren	Sean Zhou
Yaron Ukrainitz	Ning Xu	Yue Zhou
F.E. Wang	Hulya Yalcin	Ying Zhu
Hongcheng Wang	Jingyu Yan	

Sponsors

BIG - Business Information Group a.s.
Camea spol. s r.o.
Casablanca INT s.r.o.
ECVision – European Research Network for Cognitive Computer Vision Systems
Microsoft Research
Miracle Network s.r.o.
Neovision s.r.o.
Toyota

Table of Contents – Part I

Tracking I

A Unified Algebraic Approach to 2-D and 3-D Motion Segmentation	1
<i>René Vidal, Yi Ma</i>	
Enhancing Particle Filters Using Local Likelihood Sampling	16
<i>Péter Torma, Csaba Szepesvári</i>	
A Boosted Particle Filter: Multitarget Detection and Tracking	28
<i>Kenji Okuma, Ali Taleghani, Nando de Freitas, James J. Little, David G. Lowe</i>	

Feature-Based Object Detection and Recognition I

Simultaneous Object Recognition and Segmentation by Image Exploration	40
<i>Vittorio Ferrari, Tinne Tuytelaars, Luc Van Gool</i>	
Recognition by Probabilistic Hypothesis Construction	55
<i>Pierre Moreels, Michael Maire, Pietro Perona</i>	
Human Detection Based on a Probabilistic Assembly of Robust Part Detectors	69
<i>Krystian Mikolajczyk, Cordelia Schmid, Andrew Zisserman</i>	

Posters I

Model Selection for Range Segmentation of Curved Objects	83
<i>Alireza Bab-Hadiashar, Niloofar Gheissari</i>	
High-Contrast Color-Stripe Pattern for Rapid Structured-Light Range Imaging	95
<i>Changsoo Je, Sang Wook Lee, Rae-Hong Park</i>	
Using Inter-feature-Line Consistencies for Sequence-Based Object Recognition	108
<i>Jiun-Hung Chen, Chu-Song Chen</i>	
Discriminant Analysis on Embedded Manifold	121
<i>Shuicheng Yan, Hongjiang Zhang, Yuxiao Hu, Benyu Zhang, Qiansheng Cheng</i>	

Multiscale Inverse Compositional Alignment for Subdivision	
Surface Maps	133
<i>Igor Guskov</i>	
A Fourier Theory for Cast Shadows	146
<i>Ravi Ramamoorthi, Melissa Koudelka, Peter Belhumeur</i>	
Surface Reconstruction by Propagating 3D Stereo Data in	
Multiple 2D Images	163
<i>Gang Zeng, Sylvain Paris, Long Quan, Maxime Lhuillier</i>	
Visibility Analysis and Sensor Planning in Dynamic Environments	175
<i>Anurag Mittal, Larry S. Davis</i>	
Camera Calibration from the Quasi-affine Invariance of	
Two Parallel Circles	190
<i>Yihong Wu, Haijiang Zhu, Zhanyi Hu, Fuchao Wu</i>	
Texton Correlation for Recognition	203
<i>Thomas Leung</i>	
Multiple View Feature Descriptors from Image Sequences via Kernel	
Principal Component Analysis	215
<i>Jason Meltzer, Ming-Hsuan Yang, Rakesh Gupta, Stefano Soatto</i>	
An Affine Invariant Salient Region Detector	228
<i>Timor Kadir, Andrew Zisserman, Michael Brady</i>	
A Visual Category Filter for Google Images	242
<i>Robert Fergus, Pietro Perona, Andrew Zisserman</i>	
Scene and Motion Reconstruction from Defocused and Motion-Blurred	
Images via Anisotropic Diffusion	257
<i>Paolo Favaro, Martin Burger, Stefano Soatto</i>	
Semantics Discovery for Image Indexing	270
<i>Joo-Hwee Lim, Jesse S. Jin</i>	
Hand Gesture Recognition within a Linguistics-Based Framework	282
<i>Konstantinos G. Derpanis, Richard P. Wildes, John K. Tsotsos</i>	
Line Geometry for 3D Shape Understanding and Reconstruction	297
<i>Helmut Pottmann, Michael Hofer, Boris Odehnal, Johannes Wallner</i>	
Extending Interrupted Feature Point Tracking for	
3-D Affine Reconstruction	310
<i>Yasuyuki Sugaya, Kenichi Kanatani</i>	

Many-to-Many Feature Matching Using Spherical Coding of Directed Graphs	322
<i>M. Fatih Demirci, Ali Shokoufandeh, Sven Dickinson, Yakov Keselman, Lars Bretzner</i>	
Coupled-Contour Tracking through Non-orthogonal Projections and Fusion for Echocardiography	336
<i>Xiang Sean Zhou, Dorin Comaniciu, Sriram Krishnan</i>	
A Statistical Model for General Contextual Object Recognition	350
<i>Peter Carbonetto, Nando de Freitas, Kobus Barnard</i>	
Reconstruction from Projections Using Grassmann Tensors	363
<i>Richard I. Hartley, Fred Schaffalitzky</i>	
Co-operative Multi-target Tracking and Classification	376
<i>Pankaj Kumar, Surendra Ranganath, Kuntal Sengupta, Huang Weimin</i>	
A Linguistic Feature Vector for the Visual Interpretation of Sign Language	390
<i>Richard Bowden, David Windridge, Timor Kadir, Andrew Zisserman, Michael Brady</i>	
Fast Object Detection with Occlusions	402
<i>Yen-Yu Lin, Tyng-Luh Liu, Chiou-Shann Fuh</i>	
Pose Estimation of Free-Form Objects	414
<i>Bodo Rosenhahn, Gerald Sommer</i>	
Interactive Image Segmentation Using an Adaptive GMMRF Model	428
<i>Andrew Blake, Carsten Rother, M. Brown, Patrick Perez, Philip Torr</i>	
Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model	442
<i>Xianghua Ying, Zhanyi Hu</i>	
Image Clustering with Metric, Local Linear Structure, and Affine Symmetry	456
<i>Jongwoo Lim, Jeffrey Ho, Ming-Hsuan Yang, Kuang-chih Lee, David Kriegman</i>	
Face Recognition with Local Binary Patterns	469
<i>Timo Ahonen, Abdenour Hadid, Matti Pietikäinen</i>	
Steering in Scale Space to Optimally Detect Image Structures	482
<i>Jeffrey Ng, Anil A. Bharath</i>	
Hand Motion from 3D Point Trajectories and a Smooth Surface Model	495
<i>Guillaume Dewaele, Frédéric Devernay, Radu Horaud</i>	

A Robust Probabilistic Estimation Framework for Parametric Image Models	508
<i>Maneesh Singh, Himanshu Arora, Narendra Ahuja</i>	
Keyframe Selection for Camera Motion and Structure Estimation from Multiple Views	523
<i>Thorsten Thormählen, Hellward Broszio, Axel Weissenfeld</i>	
Omnidirectional Vision: Unified Model Using Conformal Geometry	536
<i>Eduardo Bayro-Corrochano, Carlos López-Franco</i>	
A Robust Algorithm for Characterizing Anisotropic Local Structures	549
<i>Kazunori Okada, Dorin Comaniciu, Navneet Dalal, Arun Krishnan</i>	
Dimensionality Reduction by Canonical Contextual Correlation Projections	562
<i>Marco Loog, Bram van Ginneken, Robert P.W. Duin</i>	
Illumination, Reflectance, and Reflection	
Accuracy of Spherical Harmonic Approximations for Images of Lambertian Objects under Far and Near Lighting	574
<i>Darya Frolova, Denis Simakov, Ronen Basri</i>	
Characterization of Human Faces under Illumination Variations Using Rank, Integrability, and Symmetry Constraints	588
<i>S. Kevin Zhou, Rama Chellappa, David W. Jacobs</i>	
User Assisted Separation of Reflections from a Single Image Using a Sparsity Prior	602
<i>Anat Levin, Yair Weiss</i>	
The Quality of Catadioptric Imaging – Application to Omnidirectional Stereo	614
<i>Wolfgang Stürzl, Hansjürgen Dahmen, Hanspeter A. Mallot</i>	
Author Index	629

Table of Contents – Part II

Geometry

A Generic Concept for Camera Calibration	1
<i>Peter Sturm, Srikumar Ramalingam</i>	
General Linear Cameras	14
<i>Jingyi Yu, Leonard McMillan</i>	
A Framework for Pencil-of-Points Structure-from-Motion	28
<i>Adrien Bartoli, Mathieu Coquerelle, Peter Sturm</i>	
What Do Four Points in Two Calibrated Images Tell Us about the Epipoles?	41
<i>David Nistér, Frederik Schaffalitzky</i>	

Feature-Based Object Detection and Recognition II

Dynamic Visual Search Using Inner-Scene Similarity: Algorithms and Inherent Limitations	58
<i>Tamar Avraham, Michael Lindenbaum</i>	
Weak Hypotheses and Boosting for Generic Object Detection and Recognition	71
<i>A. Opelt, M. Fussenegger, A. Pinz, P. Auer</i>	
Object Level Grouping for Video Shots	85
<i>Josef Sivic, Frederik Schaffalitzky, Andrew Zisserman</i>	

Posters II

Statistical Symmetric Shape from Shading for 3D Structure Recovery of Faces	99
<i>Roman Dvovard, Ronen Basri</i>	
Region-Based Segmentation on Evolving Surfaces with Application to 3D Reconstruction of Shape and Piecewise Constant Radiance	114
<i>Hailin Jin, Anthony J. Yezzi, Stefano Soatto</i>	
Human Upper Body Pose Estimation in Static Images	126
<i>Mun Wai Lee, Isaac Cohen</i>	
Automated Optic Disc Localization and Contour Detection Using Ellipse Fitting and Wavelet Transform	139
<i>P.M.D.S. Pallawala, Wynne Hsu, Mong Li Lee, Kah-Guan Au Eong</i>	

View-Invariant Recognition Using Corresponding Object Fragments	152
<i>Evgeniy Bart, Evgeny Byvatov, Shimon Ullman</i>	
Variational Pairing of Image Segmentation and Blind Restoration	166
<i>Leah Bar, Nir Sochen, Nahum Kiryati</i>	
Towards Intelligent Mission Profiles of Micro Air Vehicles:	
Multiscale Viterbi Classification	178
<i>Sinisa Todorovic, Michael C. Nechyba</i>	
Stitching and Reconstruction of Linear-Pushbroom Panoramic Images for Planar Scenes	190
<i>Chu-Song Chen, Yu-Ting Chen, Fay Huang</i>	
Audio-Video Integration for Background Modelling	202
<i>Marco Cristani, Manuele Bicego, Vittorio Murino</i>	
A Combined PDE and Texture Synthesis Approach to Inpainting	214
<i>Harald Grossauer</i>	
Face Recognition from Facial Surface Metric	225
<i>Alexander M. Bronstein, Michael M. Bronstein, Alon Spira, Ron Kimmel</i>	
Image and Video Segmentation by Anisotropic Kernel Mean Shift	238
<i>Jue Wang, Bo Thiesson, Yingqing Xu, Michael Cohen</i>	
Colour Texture Segmentation by Region-Boundary Cooperation	250
<i>Jordi Freixenet, Xavier Muñoz, Joan Martí, Xavier Lladó</i>	
Spectral Solution of Large-Scale Extrinsic Camera Calibration as a Graph Embedding Problem	262
<i>Matthew Brand, Matthew Antone, Seth Teller</i>	
Estimating Intrinsic Images from Image Sequences with Biased Illumination	274
<i>Yasuyuki Matsushita, Stephen Lin, Sing Bing Kang, Heung-Yeung Shum</i>	
Structure and Motion from Images of Smooth Textureless Objects	287
<i>Yasutaka Furukawa, Amit Sethi, Jean Ponce, David Kriegman</i>	
Automatic Non-rigid 3D Modeling from Video	299
<i>Lorenzo Torresani, Aaron Hertzmann</i>	
From a 2D Shape to a String Structure Using the Symmetry Set	313
<i>Arjan Kuijper, Ole Fogh Olsen, Peter Giblin, Philip Bille, Mads Nielsen</i>	

Extrinsic Camera Parameter Recovery from Multiple Image Sequences Captured by an Omni-directional Multi-camera System	326
<i>Tomokazu Sato, Sei Ikeda, Naokazu Yokoya</i>	
Evaluation of Robust Fitting Based Detection	341
<i>Sio-Song Ieng, Jean-Philippe Tarel, Pierre Charbonnier</i>	
Local Orientation Smoothness Prior for Vascular Segmentation of Angiography	353
<i>Wilbur C.K. Wong, Albert C.S. Chung, Simon C.H. Yu</i>	
Weighted Minimal Hypersurfaces and Their Applications in Computer Vision	366
<i>Bastian Goldlücke, Marcus Magnor</i>	
Interpolating Novel Views from Image Sequences by Probabilistic Depth Carving	379
<i>Annie Yao, Andrew Calway</i>	
Sparse Finite Elements for Geodesic Contours with Level-Sets	391
<i>Martin Weber, Andrew Blake, Roberto Cipolla</i>	
Hierarchical Implicit Surface Joint Limits to Constrain Video-Based Motion Capture	405
<i>Lorna Herda, Raquel Urtasun, Pascal Fua</i>	
Separating Specular, Diffuse, and Subsurface Scattering Reflectances from Photometric Images	419
<i>Tai-Pang Wu, Chi-Keung Tang</i>	
Temporal Factorization vs. Spatial Factorization	434
<i>Lihi Zelnik-Manor, Michal Irani</i>	
Tracking Aspects of the Foreground against the Background	446
<i>Hieu T. Nguyen, Arnold Smeulders</i>	
Example-Based Stereo with General BRDFs	457
<i>Adrien Treuille, Aaron Hertzmann, Steven M. Seitz</i>	
Adaptive Probabilistic Visual Tracking with Incremental Subspace Update	470
<i>David Ross, Jongwoo Lim, Ming-Hsuan Yang</i>	
On Refractive Optical Flow	483
<i>Sameer Agarwal, Satya P. Mallick, David Kriegman, Serge Belongie</i>	
Matching Tensors for Automatic Correspondence and Registration	495
<i>Ajmal S. Mian, Mohammed Bennamoun, Robyn Owens</i>	

A Biologically Motivated and Computationally Tractable Model of Low and Mid-Level Vision Tasks	506
<i>Iasonas Kokkinos, Rachid Deriche, Petros Maragos, Olivier Faugeras</i>	
Appearance Based Qualitative Image Description for Object Class Recognition	518
<i>Johan Thureson, Stefan Carlsson</i>	
Consistency Conditions on the Medial Axis	530
<i>Anthony Pollitt, Peter Giblin, Benjamin Kimia</i>	
Normalized Cross-Correlation for Spherical Images	542
<i>Lorenzo Sorigi, Kostas Daniilidis</i>	
Bias in the Localization of Curved Edges	554
<i>Paulo R.S. Mendonça, Dirk Padfield, James Miller, Matt Turek</i>	
Texture	
Texture Boundary Detection for Real-Time Tracking	566
<i>Ali Shahrokni, Tom Drummond, Pascal Fua</i>	
A TV Flow Based Local Scale Measure for Texture Discrimination	578
<i>Thomas Brox, Joachim Weickert</i>	
Spatially Homogeneous Dynamic Textures	591
<i>Gianfranco Doretto, Eagle Jones, Stefano Soatto</i>	
Synthesizing Dynamic Texture with Closed-Loop Linear Dynamic System	603
<i>Lu Yuan, Fang Wen, Ce Liu, Heung-Yeung Shum</i>	
Author Index	617

Table of Contents – Part III

Learning and Recognition

A Constrained Semi-supervised Learning Approach to Data Association	1
<i>Hendrik Kück, Peter Carbonetto, Nando de Freitas</i>	
Learning Mixtures of Weighted Tree-Unions by Minimizing Description Length	13
<i>Andrea Torsello, Edwin R. Hancock</i>	
Decision Theoretic Modeling of Human Facial Displays	26
<i>Jesse Hoey, James J. Little</i>	
Kernel Feature Selection with Side Data Using a Spectral Approach	39
<i>Amnon Shashua, Lior Wolf</i>	

Tracking II

Tracking Articulated Motion Using a Mixture of Autoregressive Models	54
<i>Ankur Agarwal, Bill Triggs</i>	
Novel Skeletal Representation for Articulated Creatures	66
<i>Gabriel J. Brostow, Irfan Essa, Drew Steedly, Vivek Kwatra</i>	
An Accuracy Certified Augmented Reality System for Therapy Guidance	79
<i>Stéphane Nicolau, Xavier Pennec, Luc Soler, Nichlas Ayache</i>	

Posters III

3D Human Body Tracking Using Deterministic Temporal Motion Models	92
<i>Raquel Urtasun, Pascal Fua</i>	
Robust Fitting by Adaptive-Scale Residual Consensus	107
<i>Hanzi Wang, David Suter</i>	
Causal Camera Motion Estimation by Condensation and Robust Statistics Distance Measures	119
<i>Tal Nir, Alfred M. Bruckstein</i>	

An Adaptive Window Approach for Image Smoothing and Structures Preserving	132
<i>Charles Kervrann</i>	
Extraction of Semantic Dynamic Content from Videos with Probabilistic Motion Models	145
<i>Gwenaëlle Piriou, Patrick Bouthemy, Jian-Feng Yao</i>	
Are Iterations and Curvature Useful for Tensor Voting?	158
<i>Sylvain Fischer, Pierre Bayerl, Heiko Neumann, Gabriel Cristóbal, Rafael Redondo</i>	
A Feature-Based Approach for Determining Dense Long Range Correspondences	170
<i>Josh Wills, Serge Belongie</i>	
Combining Geometric- and View-Based Approaches for Articulated Pose Estimation	183
<i>David Demirdjian</i>	
Shape Matching and Recognition – Using Generative Models and Informative Features	195
<i>Zhuowen Tu, Alan L. Yuille</i>	
Generalized Histogram: Empirical Optimization of Low Dimensional Features for Image Matching	210
<i>Shin'ichi Satoh</i>	
Recognizing Objects in Range Data Using Regional Point Descriptors . . .	224
<i>Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, Jitendra Malik</i>	
Shape Reconstruction from 3D and 2D Data Using PDE-Based Deformable Surfaces	238
<i>Ye Duan, Liu Yang, Hong Qin, Dimitris Samaras</i>	
Structure and Motion Problems for Multiple Rigidly Moving Cameras . . .	252
<i>Henrik Stewenius, Kalle Åström</i>	
Detection and Tracking Scheme for Line Scratch Removal in an Image Sequence	264
<i>Bernard Besserer, Cedric Thiré</i>	
Color Constancy Using Local Color Shifts	276
<i>Marc Ebner</i>	
Image Anisotropic Diffusion Based on Gradient Vector Flow Fields	288
<i>Hongchuan Yu, Chin-Seng Chua</i>	

Optimal Importance Sampling for Tracking in Image Sequences: Application to Point Tracking	302
<i>Elise Arnaud, Etienne Mémín</i>	
Learning to Segment	315
<i>Eran Borenstein, Shimon Ullman</i>	
MCMC-Based Multiview Reconstruction of Piecewise Smooth Subdivision Curves with a Variable Number of Control Points	329
<i>Michael Kaess, Rafal Zboinski, Frank Dellaert</i>	
Bayesian Correction of Image Intensity with Spatial Consideration	342
<i>Jiaya Jia, Jian Sun, Chi-Keung Tang, Heung-Yeung Shum</i>	
Stretching Bayesian Learning in the Relevance Feedback of Image Retrieval	355
<i>Ruofei Zhang, Zhongfei (Mark) Zhang</i>	
Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera	368
<i>Antonis A. Argyros, Manolis I.A. Lourakis</i>	
Evaluation of Image Fusion Performance with Visible Differences	380
<i>Vladimir Petrović, Costas Xydeas</i>	
An Information-Based Measure for Grouping Quality	392
<i>Erik A. Engbers, Michael Lindenbaum, Arnold W.M. Smeulders</i>	
Bias in Shape Estimation	405
<i>Hui Ji, Cornelia Fermüller</i>	
Contrast Marginalised Gradient Template Matching	417
<i>Saleh Basalamah, Anil Bharath, Donald McRobbie</i>	
The Kullback-Leibler Kernel as a Framework for Discriminant and Localized Representations for Visual Recognition	430
<i>Nuno Vasconcelos, Purdy Ho, Pedro Moreno</i>	
Partial Object Matching with Shapeme Histograms	442
<i>Y. Shan, H.S. Sawhney, B. Matei, R. Kumar</i>	
Modeling and Synthesis of Facial Motion Driven by Speech	456
<i>Payam Saisan, Alessandro Bissacco, Alessandro Chiuso, Stefano Soatto</i>	
Recovering Local Shape of a Mirror Surface from Reflection of a Regular Grid	468
<i>Silvio Savarese, Min Chen, Pietro Perona</i>	

Structure of Applicable Surfaces from Single Views	482
<i>Nail Gumerov, Ali Zandifar, Ramani Duraiswami, Larry S. Davis</i>	
Joint Bayes Filter: A Hybrid Tracker for Non-rigid Hand Motion Recognition	497
<i>Huang Fei, Ian Reid</i>	
Iso-disparity Surfaces for General Stereo Configurations	509
<i>Marc Pollefeys, Sudipta Sinha</i>	
Camera Calibration with Two Arbitrary Coplanar Circles	521
<i>Qian Chen, Haiyuan Wu, Toshikazu Wada</i>	
Reconstruction of 3-D Symmetric Curves from Perspective Images without Discrete Features	533
<i>Wei Hong, Yi Ma, Yizhou Yu</i>	
A Topology Preserving Non-rigid Registration Method Using a Symmetric Similarity Function-Application to 3-D Brain Images	546
<i>Vincent Noblet, Christian Heinrich, Fabrice Heitz, Jean-Paul Armspach</i>	
A Correlation-Based Approach to Robust Point Set Registration	558
<i>Yanghai Tsin, Takeo Kanade</i>	
Hierarchical Organization of Shapes for Efficient Retrieval	570
<i>Shantanu Joshi, Anuj Srivastava, Washington Mio, Xiuwen Liu</i>	
Information-Based Image Processing	
Intrinsic Images by Entropy Minimization	582
<i>Graham D. Finlayson, Mark S. Drew, Cheng Lu</i>	
Image Similarity Using Mutual Information of Regions	596
<i>Daniel B. Russakoff, Carlo Tomasi, Torsten Rohlfing, Calvin R. Maurer, Jr.</i>	
Author Index	609

Table of Contents – Part IV

Scale Space, Flow, Restoration

A l^1 -Unified Variational Framework for Image Restoration	1
<i>Julien Bect, Laure Blanc-Féraud, Gilles Aubert, Antonin Chambolle</i>	
Support Blob Machines. The Sparsification of Linear Scale Space	14
<i>Marco Loog</i>	
High Accuracy Optical Flow Estimation Based on a Theory for Warping	25
<i>Thomas Brox, Andrés Bruhn, Nils Papenberg, Joachim Weickert</i>	
Model-Based Approach to Tomographic Reconstruction Including Projection Deblurring. Sensitivity of Parameter Model to Noise on Data	37
<i>Jean Michel Lagrange, Isabelle Abraham</i>	

2D Shape Detection and Recognition

Unlevel-Sets: Geometry and Prior-Based Segmentation.....	50
<i>Tammy Riklin-Raviv, Nahum Kiryati, Nir Sochen</i>	
Learning and Bayesian Shape Extraction for Object Recognition	62
<i>Washington Mio, Anuj Srivastava, Xiuwen Liu</i>	
Multiphase Dynamic Labeling for Variational Recognition-Driven Image Segmentation	74
<i>Daniel Cremers, Nir Sochen, Christoph Schnörr</i>	

Posters IV

Integral Invariant Signatures	87
<i>Siddharth Manay, Byung-Woo Hong, Anthony J. Yezzi, Stefano Soatto</i>	
Detecting Keypoints with Stable Position, Orientation, and Scale under Illumination Changes	100
<i>Bill Triggs</i>	
Spectral Simplification of Graphs	114
<i>Huaijun Qiu, Edwin R. Hancock</i>	
Inferring White Matter Geometry from Diffusion Tensor MRI: Application to Connectivity Mapping.....	127
<i>Christophe Lenglet, Rachid Deriche, Olivier Faugeras</i>	

Unifying Approaches and Removing Unrealistic Assumptions in Shape from Shading: Mathematics Can Help	141
<i>Emmanuel Prados, Olivier Faugeras</i>	
Morphological Operations on Matrix-Valued Images	155
<i>Bernhard Burgeth, Martin Welk, Christian Feddern, Joachim Weickert</i>	
Constraints on Coplanar Moving Points	168
<i>Sujit Kuthirummal, C.V. Jawahar, P.J. Narayanan</i>	
A PDE Solution of Brownian Warping	180
<i>Mads Nielsen, P. Johansen</i>	
Stereovision-Based Head Tracking Using Color and Ellipse Fitting in a Particle Filter	192
<i>Bogdan Kwolek</i>	
Parallel Variational Motion Estimation by Domain Decomposition and Cluster Computing	205
<i>Timo Kohlberger, Christoph Schnörr, Andrés Bruhn, Joachim Weickert</i>	
Whitening for Photometric Comparison of Smooth Surfaces under Varying Illumination	217
<i>Margarita Osadchy, Michael Lindenbaum, David Jacobs</i>	
Structure from Motion of Parallel Lines	229
<i>Patrick Baker, Yiannis Aloimonos</i>	
A Bayesian Framework for Multi-cue 3D Object Tracking	241
<i>Jan Giebel, Darin M. Gavrilu, Christoph Schnörr</i>	
On the Significance of Real-World Conditions for Material Classification	253
<i>Eric Hayman, Barbara Caputo, Mario Fritz, Jan-Olof Eklundh</i>	
Toward Accurate Segmentation of the LV Myocardium and Chamber for Volumes Estimation in Gated SPECT Sequences	267
<i>Diane Lingrand, Arnaud Charnoz, Pierre Malick Koulibaly, Jacques Darcourt, Johan Montagnat</i>	
An MCMC-Based Particle Filter for Tracking Multiple Interacting Targets	279
<i>Zia Khan, Tucker Balch, Frank Dellaert</i>	
Human Pose Estimation Using Learnt Probabilistic Region Similarities and Partial Configurations	291
<i>Timothy J. Roberts, Stephen J. McKenna, Ian W. Ricketts</i>	

Tensor Field Segmentation Using Region Based Active Contour Model	304
<i>Zhizhou Wang, Baba C. Vemuri</i>	
Groupwise Diffeomorphic Non-rigid Registration for Automatic Model Building	316
<i>T.F. Cootes, S. Marsland, C.J. Twining, K. Smith, C.J. Taylor</i>	
Separating Transparent Layers through Layer Information Exchange	328
<i>Bernard Sarel, Michal Irani</i>	
Multiple Classifier System Approach to Model Pruning in Object Recognition.....	342
<i>Josef Kittler, Ali R. Ahmadyfard</i>	
Coaxial Omnidirectional Stereopsis	354
<i>Libor Spacek</i>	
Classifying Materials from Their Reflectance Properties	366
<i>Peter Nillius, Jan-Olof Eklundh</i>	
Seamless Image Stitching in the Gradient Domain	377
<i>Anat Levin, Assaf Zomet, Shmuel Peleg, Yair Weiss</i>	
Spectral Clustering for Robust Motion Segmentation	390
<i>JinHyeon Park, Hongyuan Zha, Rangachar Kasturi</i>	
Learning Outdoor Color Classification from Just One Training Image	402
<i>Roberto Manduchi</i>	
A Polynomial-Time Metric for Attributed Trees	414
<i>Andrea Torsello, Džena Hidović, Marcello Pelillo</i>	
Probabilistic Multi-view Correspondence in a Distributed Setting with No Central Server	428
<i>Shai Avidan, Yael Moses, Yoram Moses</i>	
Monocular 3D Reconstruction of Human Motion in Long Action Sequences	442
<i>Gareth Loy, Martin Eriksson, Josephine Sullivan, Stefan Carlsson</i>	
Fusion of Infrared and Visible Images for Face Recognition	456
<i>Aglika Gyaourova, George Bebis, Ioannis Pavlidis</i>	
Reliable Fiducial Detection in Natural Scenes	469
<i>David Claus, Andrew W. Fitzgibbon</i>	
Light Field Appearance Manifolds	481
<i>Chris Mario Christoudias, Louis-Philippe Morency, Trevor Darrell</i>	

Galilean Differential Geometry of Moving Images	494
<i>Daniel Fagerström</i>	
Tracking People with a Sparse Network of Bearing Sensors	507
<i>A. Rahimi, B. Dunagan, T. Darrell</i>	
Transformation-Invariant Embedding for Image Analysis	519
<i>Ali Ghodsi, Jiayuan Huang, Dale Schuurmans</i>	
The Least-Squares Error for Structure from Infinitesimal Motion	531
<i>John Oliensis</i>	
Stereo Based 3D Tracking and Scene Learning, Employing Particle Filtering within EM	546
<i>Trausti Kristjansson, Hagai Attias, John Hershey</i>	
3D Shape Representation and Reconstruction	
The Isophotic Metric and Its Application to Feature Sensitive Morphology on Surfaces	560
<i>Helmut Pottmann, Tibor Steiner, Michael Hofer, Christoph Haider, Allan Hanbury</i>	
A Closed-Form Solution to Non-rigid Shape and Motion Recovery	573
<i>Jing Xiao, Jin-xiang Chai, Takeo Kanade</i>	
Stereo Using Monocular Cues within the Tensor Voting Framework	588
<i>Philippos Mordohai, Gérard Medioni</i>	
Shape and View Independent Reflectance Map from Multiple Views	602
<i>Tianli Yu, Ning Xu, Narendra Ahuja</i>	
Author Index	617

A Unified Algebraic Approach to 2-D and 3-D Motion Segmentation^{*}

René Vidal^{1,2} and Yi Ma³

¹ Center for Imaging Science, Johns Hopkins University, Baltimore MD 21218, USA

² National ICT Australia, Canberra ACT 0200, Australia, rvidal@cis.jhu.edu

³ Dept. of Elect. and Comp. Eng., UIUC, Urbana, IL 61801, USA, yima@uiuc.edu

Abstract. We present an analytic solution to the problem of estimating multiple 2-D and 3-D motion models from two-view correspondences or optical flow. The key to our approach is to view the estimation of multiple motion models as the estimation of a single *multibody motion model*. This is possible thanks to two important algebraic facts. First, we show that all the image measurements, regardless of their associated motion model, can be *fit* with a real or complex *polynomial*. Second, we show that the parameters of the motion model associated with an image measurement can be obtained from the *derivatives* of the polynomial at the measurement. This leads to a novel motion segmentation algorithm that applies to most of the two-view motion models adopted in computer vision. Our experiments show that the proposed algorithm outperforms existing algebraic methods in terms of efficiency and robustness, and provides a good initialization for iterative techniques, such as EM, which is strongly dependent on correct initialization.

1 Introduction

A classic problem in visual motion analysis is to estimate a motion model for a set of 2-D feature points as they move in a video sequence. Ideally, one would like to fit a single model that describes the motion of all the features. In practice, however, different regions of the image obey different motion models due to depth discontinuities, perspective effects, multiple moving objects, etc. Therefore, one is faced with the problem of fitting multiple motion models to the image, without knowing which pixels are moving according to the same model. More specifically:

Problem 1 (Multiple-motion estimation and segmentation). Given a set of image measurements $\{(\mathbf{x}_1^j, \mathbf{x}_2^j)\}_{j=1}^N$ taken from two views of a motion sequence related by a collection of n (n known) 2-D or 3-D motion models $\{\mathcal{M}_i\}_{i=1}^n$, estimate the motion models without knowing which image measurements correspond to which motion model.

Related literature. There is a rich literature addressing the 2-D motion segmentation problem using the so-called layered representation [1] or different variations of the Expectation Maximization (EM) algorithm [2,3,4]. These approaches alternate between the segmentation of the image measurements (E-step) and the estimation of the motion

^{*} The authors thank Jacopo Piazzi and Frederik Schaffalitzky for fruitful discussions. Research funded with startup funds from the departments of BME at Johns Hopkins and ECE at UIUC.

parameters (M-step) and suffer from the disadvantage that the convergence to the optimal solution strongly depends on correct initialization [5,6]. Existing initialization techniques estimate the motion parameters from local patches and cluster these motion parameters using K-means [7], normalized cuts [5], or a Bayesian version of RANSAC [6]. The only existing algebraic solution to 2-D motion segmentation is based on bi-homogeneous polynomial factorization and can be found in [9].

The 3-D motion segmentation problem has received relatively less attention. Existing approaches include combinations of EM with normalized cuts [8] and factorization methods for orthographic and affine cameras [10,11]. Algebraic approaches based on polynomial and tensor factorization have been proposed in the case of multiple translating objects [12] and in the case of two [13] and multiple [14] rigid-body motions.

Our contribution. In this paper, we address the initialization of iterative approaches to motion estimation and segmentation by proposing a *non-iterative* algebraic solution to Problem 1 that applies to most 2-D and 3-D motion models in computer vision, as detailed in Table 1. The key to our approach is to view the estimation of multiple motion models as the estimation of a *single*, though more complex, *multibody motion model* that is then factored into the original models. This is achieved by (1) eliminating the feature segmentation problem in an algebraic fashion, (2) fitting a single multibody motion model to all the image measurements, and (3) segmenting the multibody motion model into its individual components. More specifically, our approach proceeds as follows:

1. *Eliminate Feature Segmentation:* Find an algebraic equation that is satisfied by all the image measurements, regardless of the motion model associated with each measurement. For the motion models considered in this paper, the i^{th} motion model will be typically defined by an algebraic equation of the form $f(\mathbf{x}_1, \mathbf{x}_2, \mathcal{M}_i) = 0$. Therefore an algebraic equation that is satisfied by all the data is

$$g(\mathbf{x}_1, \mathbf{x}_2, \mathcal{M}) = f(\mathbf{x}_1, \mathbf{x}_2, \mathcal{M}_1)f(\mathbf{x}_1, \mathbf{x}_2, \mathcal{M}_2) \cdots f(\mathbf{x}_1, \mathbf{x}_2, \mathcal{M}_n) = 0. \quad (1)$$

Such an equation represents a single *multibody motion model* whose parameters \mathcal{M} encode those of the original motion models $\{\mathcal{M}_i\}_{i=1}^n$.

2. *Multibody Motion Estimation:* Estimate the parameters \mathcal{M} of the multibody motion model from the given image measurements. For the motion models considered in this paper, the parameters \mathcal{M} will correspond to the coefficients of a real or complex polynomial p_n of degree n . We will show that if n is known such parameters can be estimated *linearly* after embedding the image data into a higher-dimensional space.
3. *Motion Segmentation:* Recover the parameters of the original motion models from the parameters of the multibody motion model \mathcal{M} , i.e.

$$\mathcal{M} \rightarrow \{\mathcal{M}_i\}_{i=1}^n. \quad (2)$$

We will show that the individual motion parameters \mathcal{M}_i can be computed from the *derivatives* of p_n evaluated at a collection of n image measurements.

This new approach offers two important technical advantages over previously known algebraic solutions to the segmentation of 3-D translational [12] and rigid-body motions (fundamental matrices) [14] based on homogeneous *polynomial factorization*:

Table 1. 2-D and 3-D motion models considered in this paper

Motion models	Model equations	Model parameters	Equivalent to clustering
2-D translational	$\mathbf{x}_2 = \mathbf{x}_1 + T_i$	$\{T_i \in \mathbb{R}^2\}_{i=1}^n$	Hyperplanes in \mathbb{C}^2
2-D similarity	$\mathbf{x}_2 = \lambda_i R_i \mathbf{x}_1 + T_i$	$\{(R_i, T_i) \in SE(2), \lambda_i \in \mathbb{R}^+\}_{i=1}^n$	Hyperplanes in \mathbb{C}^3
2-D affine	$\mathbf{x}_2 = A_i \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}$	$\{A_i \in \mathbb{R}^{2 \times 3}\}_{i=1}^n$	Hyperplanes in \mathbb{C}^4
3-D translational	$0 = \mathbf{x}_2^T [T_i]_{\times} \mathbf{x}_1$	$\{T_i \in \mathbb{R}^3\}_{i=1}^n$	Hyperplanes in \mathbb{R}^3
3-D rigid-body	$0 = \mathbf{x}_2^T F_i \mathbf{x}_1$	$\{F_i \in \mathbb{R}^{3 \times 3} : \text{rank}(F_i) = 2\}_{i=1}^n$	Bilinear forms in $\mathbb{R}^{3 \times 3}$
3-D homography	$\mathbf{x}_2 \sim H_i \mathbf{x}_1$	$\{H_i \in \mathbb{R}^{3 \times 3}\}_{i=1}^n$	Bilinear forms in $\mathbb{C}^{2 \times 3}$

1. It is based on *polynomial differentiation* rather than *polynomial factorization*, which greatly improves the efficiency, accuracy and robustness of the algorithm.
2. It applies to either feature correspondences or optical flows and includes most of the two-view motion models in computer vision: 2-D translational, similarity, and affine, or 3-D translational, rigid body motions (fundamental matrices), or motions of planar scenes (homographies), as shown in Table 1. The unification is achieved by embedding some of the motion models into the complex domain, which resolves cases such as 2-D affine motions and 3-D homographies that could not be solved in the real domain.

With respect to extant probabilistic methods, our approach has the advantage that it provides a global, non-iterative solution that does not need initialization. Therefore, our method can be used to initialize any iterative or optimization based technique, such as EM, or else in a layered (multiscale) or hierarchical fashion at the user's discretion.

Noisy image data. Although the derivation of the algorithm will assume noise free data, the algorithm is designed to work with moderate noise, as we will soon point out.

Notation. Let \mathbf{z} be a vector in \mathbb{R}^K or \mathbb{C}^K and let \mathbf{z}^T be its transpose. A homogeneous polynomial of degree n in \mathbf{z} is a polynomial $p_n(\mathbf{z})$ such that $p_n(\lambda \mathbf{z}) = \lambda^n p_n(\mathbf{z})$ for all λ in \mathbb{R} or \mathbb{C} . The space of all homogeneous polynomials of degree n in K variables, $R_n(K)$, is a vector space of dimension $M_n(K) = \binom{n+K-1}{K-1} = \binom{n+K-1}{n}$. A particular basis for $R_n(K)$ is obtained by considering all the monomials of degree n in K variables, that is $\mathbf{z}^I = z_1^{n_1} z_2^{n_2} \cdots z_K^{n_K}$ with $0 \leq n_j \leq n$ for $j = 1, \dots, K$, and $n_1 + n_2 + \cdots + n_K = n$. Therefore, each polynomial $p_n(\mathbf{z}) \in R_n(K)$ can be written as a linear combination of a vector of coefficients $\mathbf{c} \in \mathbb{R}^{M_n(K)}$ or $\mathbb{C}^{M_n(K)}$ as

$$p_n(\mathbf{z}) = \mathbf{c}^T \nu_n(\mathbf{z}) = \sum c_{n_1, n_2, \dots, n_K} z_1^{n_1} z_2^{n_2} \cdots z_K^{n_K}, \quad (3)$$

where $\nu_n: \mathbb{R}^K(\mathbb{C}^K) \rightarrow \mathbb{R}^{M_n(K)}(\mathbb{C}^{M_n(K)})$ is the *Veronese map* of degree n [12] defined as $\nu_n: [z_1, \dots, z_K]^T \mapsto [\dots, \mathbf{z}^I, \dots]^T$ with I chosen in the degree-lexicographic order. The Veronese map is also known as the *polynomial embedding* in the machine learning community.

2 2-D Motion Segmentation by Clustering Hyperplanes in \mathbb{C}^K

2.1 Segmentation of 2-D Translational Motions: Clustering Hyperplanes in \mathbb{C}^2

The case of feature points. Under the 2-D translational motion model the two images are related by one out of n possible 2-D translations $\{T_i \in \mathbb{R}^2\}_{i=1}^n$. That is, for each feature pair $\mathbf{x}_1 \in \mathbb{R}^2$ and $\mathbf{x}_2 \in \mathbb{R}^2$ there exists a 2-D translation $T_i \in \mathbb{R}^2$ such that

$$\mathbf{x}_2 = \mathbf{x}_1 + T_i. \quad (4)$$

Therefore, if we interpret the displacement of the features $(\mathbf{x}_2 - \mathbf{x}_1)$ and the 2-D translations T_i as complex numbers $(\mathbf{x}_2 - \mathbf{x}_1) \in \mathbb{C}$ and $T_i \in \mathbb{C}$, then we can re-write equation (4) as

$$\mathbf{b}_i^T \mathbf{z} \doteq [T_i \ 1] \begin{bmatrix} 1 \\ -(\mathbf{x}_2 - \mathbf{x}_1) \end{bmatrix} = 0 \in \mathbb{C}^2. \quad (5)$$

The above equation corresponds to a hyperplane in \mathbb{C}^2 whose normal vector \mathbf{b}_i encodes the 2-D translational motion T_i . Therefore, the segmentation of n 2-D translational motions $\{T_i \in \mathbb{R}^2\}_{i=1}^n$ from a set of correspondences $\{\mathbf{x}_1^j \in \mathbb{R}^2\}_{j=1}^N$ and $\{\mathbf{x}_2^j \in \mathbb{R}^2\}_{j=1}^N$ is equivalent to clustering data points $\{\mathbf{z}^j \in \mathbb{C}^2\}_{j=1}^N$ lying on n complex hyperplanes with normal vectors $\{\mathbf{b}_i \in \mathbb{C}^2\}_{i=1}^n$. As we will see in short, other 2-D and 3-D motion segmentation problems are also equivalent to clustering data lying on complex hyperplanes in \mathbb{C}^3 and \mathbb{C}^4 . Therefore, rather than solving the hyperplane clustering problem for the case $K = 2$, we now present a solution for hyperplanes in \mathbb{C}^K with arbitrary K by adapting the Generalized PCA algorithm of [15] to the complex domain.

Eliminating feature segmentation. We first notice that each point $\mathbf{z} \in \mathbb{C}^K$, regardless of which motion model $\{\mathbf{b}_i \in \mathbb{C}^K\}_{i=1}^n$ is associated with it, must satisfy the following homogeneous polynomial of degree n in K complex variables

$$p_n(\mathbf{z}) = \prod_{i=1}^n (\mathbf{b}_i^T \mathbf{z}) = \sum_I c_I \mathbf{z}^I = \sum c_{n_1, \dots, n_K} z_1^{n_1} z_2^{n_2} \cdots z_K^{n_K} = \mathbf{c}^T \boldsymbol{\nu}_n(\mathbf{z}) = 0, \quad (6)$$

where the coefficient vector $\mathbf{c} \in \mathbb{C}^{M_n(K)}$ represents the *multibody motion parameters*.

Estimating multibody motion. Since the polynomial p_n must be satisfied by all the data points $\mathbf{Z} = \{\mathbf{z}^j \in \mathbb{C}^K\}_{j=1}^N$, we obtain the following linear system on \mathbf{c}

$$\boxed{L_n \mathbf{c} = 0 \quad \in \mathbb{C}^N}, \quad (7)$$

where $L_n = [\boldsymbol{\nu}_n(\mathbf{z}^1), \boldsymbol{\nu}_n(\mathbf{z}^2), \dots, \boldsymbol{\nu}_n(\mathbf{z}^N)]^T \in \mathbb{C}^{N \times M_n(K)}$. One can show that there is a unique solution for \mathbf{c} (up to a scale factor) if $N \geq M_n(K) - 1$ and at least $K - 1$ points belong to each hyperplane. Furthermore, since the last entry of each \mathbf{b}_i is equal to one, then so is the last entry of \mathbf{c} . Therefore, one can solve for \mathbf{c} uniquely. In the presence of noise, one can solve for \mathbf{c} in a least-squares sense as the singular vector of L_n associated with its smallest singular value, and then normalize so that $\mathbf{c}_{M_n(K)} = 1$.

Segmenting the multibody motion. Given c , we now present an algorithm for computing the motion parameters \mathbf{b}_i from the derivatives of p_n . To this end, we consider the derivative of $p_n(\mathbf{z})$,

$$Dp_n(\mathbf{z}) = \frac{\partial p_n(\mathbf{z})}{\partial \mathbf{z}} = \sum_{i=1}^n \prod_{\ell \neq i} (\mathbf{b}_\ell^T \mathbf{z}) \mathbf{b}_i, \quad (8)$$

and notice that if we evaluate $Dp_n(\mathbf{z})$ at a point $\mathbf{z} = \mathbf{y}_i$ that corresponds to the i^{th} motion model, i.e. if \mathbf{y}_i is such that $\mathbf{b}_i^T \mathbf{y}_i = 0$, then we have $Dp_n(\mathbf{y}_i) \sim \mathbf{b}_i$. Therefore, given c we can obtain the motion parameters as

$$\mathbf{b}_i = \left. \frac{Dp_n(\mathbf{z})}{e_K^T Dp_n(\mathbf{z})} \right|_{\mathbf{z}=\mathbf{y}_i}, \quad (9)$$

where $e_K = [0, \dots, 0, 1]^T \in \mathbb{C}^K$ and $\mathbf{y}_i \in \mathbb{C}^K$ is a nonzero vector such that $\mathbf{b}_i^T \mathbf{y}_i = 0$.

The rest of the problem is to find one vector $\mathbf{y}_i \in \mathbb{C}^K$ in each one of the hyperplanes $\mathcal{H}_i = \{\mathbf{z} \in \mathbb{C}^K : \mathbf{b}_i^T \mathbf{z} = 0\}$ for $i = 1, \dots, n$. To this end, notice that we can always choose a point \mathbf{y}_n lying on one of the hyperplanes as any of the points in the data set \mathbf{Z} . However, in the presence of noise and outliers, an arbitrary point in \mathbf{Z} may be far from the hyperplanes. The question is then how to compute the *distance* from each data point to its closest hyperplane, *without* knowing the normals to the hyperplanes. The following lemma allows us to compute a first order approximation to such a distance:

Lemma 1. *Let $\tilde{\mathbf{z}} \in \mathcal{H}_i$ be the projection of a point $\mathbf{z} \in \mathbb{C}^K$ onto its closest hyperplane \mathcal{H}_i . Also let $\Pi = (I - e_K e_K^T)$. Then the Euclidean distance from \mathbf{z} to \mathcal{H}_i is given by*

$$\|\mathbf{z} - \tilde{\mathbf{z}}\| = \frac{|p_n(\mathbf{z})|}{\|\Pi Dp_n(\mathbf{z})\|} + O(\|\mathbf{z} - \tilde{\mathbf{z}}\|^2). \quad (10)$$

Therefore, we can choose a point in the data set close to one of the subspaces as:

$$\mathbf{y}_n = \arg \min_{\mathbf{z} \in \mathbf{Z}} \frac{|p_n(\mathbf{z})|}{\|\Pi Dp_n(\mathbf{z})\|}, \quad (11)$$

and then compute the normal vector at \mathbf{y}_n as $\mathbf{b}_n = Dp_n(\mathbf{y}_n)/(e_K^T Dp_n(\mathbf{y}_n))$. In order to find a point \mathbf{y}_{n-1} in one of the remaining hyperplanes, we could just remove the points on \mathcal{H}_n from \mathbf{Z} and compute \mathbf{y}_{n-1} similarly to (11), but minimizing over $\mathbf{Z} \setminus \mathcal{H}_n$, and so on. However, the above process is not very robust in the presence of noise. Therefore, we propose an alternative solution that penalizes choosing a point from \mathcal{H}_n in (11) by dividing the objective function by the distance from \mathbf{z} to \mathcal{H}_n , namely $|\mathbf{b}_n^T \mathbf{z}|/\|\Pi \mathbf{b}_n\|$. That is, we can choose a point on or close to $\cup_{i=1}^{n-1} \mathcal{H}_i$ as

$$\mathbf{y}_{n-1} = \arg \min_{\mathbf{z} \in \mathbf{Z}} \frac{\frac{|p_n(\mathbf{z})|}{\|\Pi Dp_n(\mathbf{z})\|} + \delta}{\frac{|\mathbf{b}_n^T \mathbf{z}|}{\|\Pi \mathbf{b}_n\|} + \delta}, \quad (12)$$

where $\delta > 0$ is a small positive number chosen to avoid cases in which both the numerator and the denominator are zero (e.g. with perfect data). By repeating this process for the remaining hyperplanes, we obtain the following hyperplane clustering algorithm:

Algorithm 1 (Clustering hyperplanes in \mathbb{C}^K) Given data points $\mathbf{Z} = \{\mathbf{z}^j \in \mathbb{C}^K\}_{j=1}^N$
solve for $\mathbf{c} \in \mathbb{C}^{M_n(K)}$ from the linear system $[\nu_n(\mathbf{z}^1), \nu_n(\mathbf{z}^2), \dots, \nu_n(\mathbf{z}^N)]^T \mathbf{c} = 0$;
set $p_n(\mathbf{z}) = \mathbf{c}^T \nu_n(\mathbf{z})$;
for $i = n : 1$,

$$\mathbf{y}_i = \arg \min_{\mathbf{z} \in \mathbf{Z}} \frac{\frac{|p_n(\mathbf{z})|}{\|Dp_n(\mathbf{z})\|} + \delta}{\frac{|\mathbf{b}_{i+1}^T \mathbf{z}| \cdots |\mathbf{b}_n^T \mathbf{z}|}{\|\Pi \mathbf{b}_{i+1}\| \cdots \|\Pi \mathbf{b}_n\|} + \delta}; \quad \mathbf{b}_i = \frac{Dp_n(\mathbf{y}_i)}{e_K^T Dp_n(\mathbf{y}_i)}; \quad (13)$$

end.

Notice that one could also choose the points \mathbf{y}_i in a purely algebraic fashion, e.g., by intersecting a random line with the hyperplanes, or else by dividing the polynomial $p_n(\mathbf{z})$ by $\mathbf{b}_n^T \mathbf{z}$. However, we have chosen to present Algorithm 1 instead, because it has a better performance with noisy data and is not very sensitive to the choice of δ .

The case of translational optical flow. Imagine now that rather than a collection of feature points we are given the optical flow $\{\mathbf{u}_j \in \mathbb{R}^2\}_{j=1}^N$ between two consecutive views of a video sequence. If we assume that the optical flow is piecewise constant, i.e. the optical flow of every pixel in the image takes only n possible values $\{T_i \in \mathbb{R}^2\}_{i=1}^n$, then at each pixel $j \in \{1, \dots, N\}$ there exists a motion T_i such that

$$\mathbf{u}_j = T_i. \quad (14)$$

The problem is now to estimate the n motion models $\{T_i\}_{i=1}^n$ from the optical flow $\{\mathbf{u}_j\}_{j=1}^N$. If $N \geq M_n(2) - 1 \sim O(n)$, this problem can be solved using the same technique as in the case of feature points (Algorithm 1 with $K = 3$) after replacing $\mathbf{x}_2 - \mathbf{x}_1 = \mathbf{u}$.

2.2 Segmentation of 2-D Similarity Motions: Clustering Hyperplanes in \mathbb{C}^3

The case of feature points. In this case, we assume that for each feature point $(\mathbf{x}_1, \mathbf{x}_2)$ there exists a 2-D rigid-body motion $(R_i, T_i) \in SE(2)$ and a scale $\lambda_i \in \mathbb{R}^+$ such that

$$\mathbf{x}_2 = \lambda_i R_i \mathbf{x}_1 + T_i = \lambda_i \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \mathbf{x}_1 + T_i. \quad (15)$$

Therefore, if we interpret the rotation matrix as a unit number $R_i = \exp(\theta_i \sqrt{-1}) \in \mathbb{C}$, and the translation vector and the image features as points in the complex plane $T_i, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{C}$, then we can write the 2-D similarity motion model as the following hyperplane in \mathbb{C}^3 :

$$\mathbf{b}_i^T \mathbf{z} \doteq [\lambda_i R_i \ T_i \ 1] \begin{bmatrix} \mathbf{x}_1 \\ 1 \\ -\mathbf{x}_2 \end{bmatrix} = 0. \quad (16)$$

Therefore, the segmentation of 2-D similarity motions is equivalent to clustering hyperplanes in \mathbb{C}^3 . As such, we can apply Algorithm 1 with $K = 3$ to a collection of

$N \geq M_n(3) - 1 \sim O(n^2)$ image measurements $\{\mathbf{z}^j \in \mathbb{C}^3\}_{j=1}^N$, with at least two measurements per motion model, to obtain the motion parameters $\{\mathbf{b}_i \in \mathbb{C}^3\}_{i=1}^n$. The original *real* motion parameters are then given as

$$\lambda_i = |\mathbf{b}_{i1}|, \quad \theta_i = \angle \mathbf{b}_{i1}, \quad \text{and} \quad T_i = [\text{Re}(\mathbf{b}_{i2}), \text{Im}(\mathbf{b}_{i2})]^T, \quad \text{for } i = 1, \dots, n. \quad (17)$$

The case of optical flow. Let $\{\mathbf{u}_j \in \mathbb{R}^2\}_{j=1}^N$ be N measurements of the optical flow at the N pixels $\{\mathbf{x}_j \in \mathbb{R}^2\}_{j=1}^N$. We assume that the optical flow field can be modeled as a collection of n 2-D similarity motion models as $\mathbf{u} = \lambda_i R_i \mathbf{x} + T_i$. Therefore, the segmentation of 2-D similarity motions from measurements of optical flow can be solved as in the case of feature points, after replacing $\mathbf{x}_2 = \mathbf{u}$ and $\mathbf{x}_1 = \mathbf{x}$.

2.3 Segmentation of 2-D Affine Motions: Clustering Hyperplanes in \mathbb{C}^4

The case of feature points. In this case, we assume that the images are related by a collection of n 2-D affine motion models $\{A_i \in \mathbb{R}^{2 \times 3}\}_{i=1}^n$. That is, for each feature pair $(\mathbf{x}_1, \mathbf{x}_2)$ there exist a 2-D affine motion A_i such that

$$\mathbf{x}_2 = A_i \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}_i \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}. \quad (18)$$

Therefore, if we interpret \mathbf{x}_2 as a complex number $\mathbf{x}_2 \in \mathbb{C}$, but we still think of \mathbf{x}_1 as a vector in \mathbb{R}^2 , then we have

$$\mathbf{x}_2 = \mathbf{a}_i^T \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix} = [a_{11} + a_{21}\sqrt{-1} \quad a_{12} + a_{22}\sqrt{-1} \quad a_{13} + a_{23}\sqrt{-1}]_i \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}. \quad (19)$$

The above equation represents the following hyperplane in \mathbb{C}^4

$$\mathbf{b}_i^T \mathbf{z} = [\mathbf{a}_i^T \quad 1] \begin{bmatrix} \mathbf{x}_1 \\ 1 \\ -\mathbf{x}_2 \end{bmatrix} = 0, \quad (20)$$

where the normal vector $\mathbf{b}_i \in \mathbb{C}^4$ encodes the affine motion parameters and the data point $\mathbf{z} \in \mathbb{C}^4$ encodes the image measurements $\mathbf{x}_1 \in \mathbb{R}^2$ and $\mathbf{x}_2 \in \mathbb{C}$. Therefore, the segmentation of 2-D affine motion models is equivalent to clustering hyperplanes in \mathbb{C}^4 . As such, we can apply Algorithm 1 with $K = 4$ to a collection of $N \geq M_n(4) - 1 \sim O(n^3)$ image measurements $\{\mathbf{z}^j \in \mathbb{C}^4\}_{j=1}^N$, with at least three measurements per motion model, to obtain the motion parameters $\{\mathbf{b}_i \in \mathbb{C}^3\}_{i=1}^n$. The original affine motion models are then obtained as

$$A_i = \begin{bmatrix} \text{Re}(\mathbf{b}_{i1}) & \text{Re}(\mathbf{b}_{i2}) & \text{Re}(\mathbf{b}_{i3}) \\ \text{Im}(\mathbf{b}_{i1}) & \text{Im}(\mathbf{b}_{i2}) & \text{Im}(\mathbf{b}_{i3}) \end{bmatrix} \in \mathbb{R}^{2 \times 3}, \quad \text{for } i = 1, \dots, n. \quad (21)$$

The case of affine optical flow. In this case, the optical flow \mathbf{u} is modeled as being generated by a collection of n affine motion models $\{A_i \in \mathbb{R}^{2 \times 3}\}_{i=1}^n$ of the form $\mathbf{u} = A_i \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$. Therefore, the segmentation of 2-D affine motions can be solved as in the case of feature points, after replacing $\mathbf{x}_2 = \mathbf{u}$ and $\mathbf{x}_1 = \mathbf{x}$.

3 3-D Motion Segmentation

3.1 Segmentation of 3-D Translational Motions: Clustering Hyperplanes in \mathbb{R}^3

The case of feature points. In this case, we assume that the scene can be modeled as a mixture of purely translational motion models, $\{T_i \in \mathbb{R}^3\}_{i=1}^n$, where T_i represents the translation (calibrated case) or the *epipole* (uncalibrated case) of object i relative to the camera between the two frames. A solution to this problem based on polynomial factorization was proposed in [12]. Here we present a much simpler solution based on polynomial differentiation.

Given the images $\mathbf{x}_1 \in \mathbb{P}^2$ and $\mathbf{x}_2 \in \mathbb{P}^2$ of a point in object i in the first and second frame, they must satisfy the well-known epipolar constraint for linear motions

$$-\mathbf{x}_2^T [T_i]_{\times} \mathbf{x}_1 = T_i^T (\mathbf{x}_2 \times \mathbf{x}_1) = T_i^T \ell = 0, \quad (22)$$

where $\ell = (\mathbf{x}_2 \times \mathbf{x}_1) \in \mathbb{R}^3$ is known as the *epipolar line* associated with the image pair $(\mathbf{x}_1, \mathbf{x}_2)$. Therefore, the segmentation of 3-D translational motions is equivalent to clustering data (epipolar lines) lying on a collection of hyperplanes in \mathbb{R}^3 whose normal vectors are the n epipoles $\{T_i\}_{i=1}^n$. As such, we can apply Algorithm 1 with $K = 3$ to $N \geq M_n(3) - 1 \sim O(n^2)$ epipolar lines $\{\ell^j = \mathbf{x}_1^j \times \mathbf{x}_2^j\}_{j=1}^N$, with at least two epipolar lines per motion, to estimate the epipoles $\{T_i\}_{i=1}^n$ from the derivatives of the polynomial $p_n(\ell) = (T_1^T \ell) \cdots (T_n^T \ell)$. The only difference is that in this case the last entry of each epipole is *not* constrained to be equal to one. Therefore, when choosing the points \mathbf{y}_i in equation (13) we should take $\Pi = I$ not to eliminate the last coordinate. We therefore compute the epipoles up to an unknown scale factor as

$$T_i = Dp_n(\mathbf{y}_i) / \|Dp_n(\mathbf{y}_i)\|, \quad i = 1, \dots, n, \quad (23)$$

where the unknown scale is lost under perspective projection.

The case of optical flow. In the case of optical flow generated by purely translating objects we have $\mathbf{u}^T [T_i]_{\times} \mathbf{x} = 0$, where \mathbf{u} is interpreted as a three vector $[\mathbf{u}, \mathbf{v}, 0]^T \in \mathbb{R}^3$. Thus, one can estimate the translations $\{T_i \in \mathbb{R}^3\}_{i=1}^n$ as before by replacing $\mathbf{x}_2 = \mathbf{u}$ and $\mathbf{x}_1 = \mathbf{x}$.

3.2 Segmentation of 3-D Rigid-Body Motions: Clustering Quadratic Forms in $\mathbb{R}^{3 \times 3}$

Assume that the motion of the objects relative to the camera between the two views can be modeled as a mixture of 3-D rigid-body motions $\{(R_i, T_i) \in SE(3)\}_{i=1}^n$ which are represented with a nonzero rank-2 *fundamental matrix* F_i . A solution to this problem based on the factorization of bi-homogeneous polynomials was proposed in [14]. Here we present a much simpler solution based on taking derivatives of the so-called multibody epipolar constraint (see below), thus avoiding polynomial factorization.

Given an image pair $(\mathbf{x}_1, \mathbf{x}_2)$, there exists a motion i such that the following epipolar constraint is satisfied

$$\mathbf{x}_2^T F_i \mathbf{x}_1 = 0. \quad (24)$$

Therefore, the following *multibody epipolar constraint* [14] must be satisfied by the number of independent motions n , the fundamental matrices $\{F_i\}_{i=1}^n$ and the image pair $(\mathbf{x}_1, \mathbf{x}_2)$, regardless of the object to which the image pair belongs

$$p_n(\mathbf{x}_1, \mathbf{x}_2) \doteq \prod_{i=1}^n (\mathbf{x}_2^T F_i \mathbf{x}_1) = 0. \quad (25)$$

It was also shown in [14] that the multibody epipolar constraint can be written in bilinear form as $\nu_n(\mathbf{x}_2)^T \mathcal{F} \nu_n(\mathbf{x}_1) = 0$, where $\mathcal{F} \in \mathbb{R}^{M_n(3) \times M_n(3)}$ is the so-called *multibody fundamental matrix*, which can be linearly estimated from $N \geq M_n(3)^2 - 1 \sim O(n^4)$ image pairs in general position with at least 8 pairs corresponding to each motion.

We now present a new solution to the problem of estimating the fundamental matrices $\{F_i\}_{i=1}^n$ from the multibody fundamental matrix \mathcal{F} based on taking derivatives of the multibody epipolar constraint. Recall that, given a point $\mathbf{x}_1 \in \mathbb{P}^2$ in the first image frame, the epipolar lines associated with it are defined as $\ell_i \doteq F_i \mathbf{x}_1 \in \mathbb{R}^3, i = 1, \dots, n$. Therefore, if the image pair $(\mathbf{x}_1, \mathbf{x}_2)$ corresponds to motion i , i.e. if $\mathbf{x}_2^T F_i \mathbf{x}_1 = 0$, then

$$\frac{\partial}{\partial \mathbf{x}_2} \nu_n(\mathbf{x}_2)^T \mathcal{F} \nu_n(\mathbf{x}_1) = \sum_{i=1}^n \prod_{\ell \neq i} (\mathbf{x}_2^T F_\ell \mathbf{x}_1) (F_i \mathbf{x}_1) = \prod_{\ell \neq i} (\mathbf{x}_2^T F_\ell \mathbf{x}_1) (F_i \mathbf{x}_1) \sim \ell_i. \quad (26)$$

In other words, the partial derivative of the multibody epipolar constraint with respect to \mathbf{x}_2 evaluated at $(\mathbf{x}_1, \mathbf{x}_2)$ is proportional to *the* epipolar line associated with $(\mathbf{x}_1, \mathbf{x}_2)$ in the second view.¹ Therefore, given a set of image pairs $\{(\mathbf{x}_1^j, \mathbf{x}_2^j)\}_{j=1}^N$ and the multibody fundamental matrix $\mathcal{F} \in \mathbb{R}^{M_n(3) \times M_n(3)}$, we can estimate a collection of epipolar lines $\{\ell^j\}_{j=1}^N$. Remember from Section 3.1 that in the case of purely translating objects the epipolar lines were readily obtained as $\mathbf{x}_1 \times \mathbf{x}_2$. Here the calculation is more involved because of the rotational component of the rigid-body motions. Nevertheless, given a set of epipolar lines we can apply Algorithm 1 with $K = 3$ and $\Pi = I$ to estimate the n epipoles $\{T_i\}_{i=1}^n$ up to a scale factor, as in equation (23). Therefore, if the n epipoles are different,² then we can immediately compute the n fundamental matrices $\{F_i\}_{i=1}^n$ by assigning the image pair $(\mathbf{x}_1^j, \mathbf{x}_2^j)$ to group i if $i = \arg \min_{\ell=1, \dots, n} (T_i^T \ell^j)^2$ and then applying the eight-point algorithm to the image pairs in group $i = 1, \dots, n$.

3.3 Segmentation of 3-D Homographies: Clustering Quadratic Forms in $\mathbb{C}^{2 \times 3}$

The motion segmentation scheme described in the previous section assumes that the displacement of each object between the two views relative to the camera is nonzero, i.e. $T_i \neq 0$, otherwise the individual fundamental matrices are zero. Furthermore, it also

¹ Similarly, the partial derivative of the multibody epipolar constraint with respect to \mathbf{x}_1 evaluated at $(\mathbf{x}_1, \mathbf{x}_2)$ is proportional to *the* epipolar line associated with $(\mathbf{x}_1, \mathbf{x}_2)$ in the first view.

² Notice that this is not a strong assumption. If two individual fundamental matrices share the same (left) epipoles, one can consider the right epipoles (in the first image frame) instead, because it is extremely rare that two motions give rise to the same left and right epipoles. In fact, this happens only when the rotation axes of the two motions are equal to each other and parallel to the translation direction [14].

requires that the 3-D points be in general configuration, otherwise one cannot uniquely recover each fundamental matrix from its epipolar constraint. The latter case occurs, for example, in the case of planar structures, i.e. when the 3-D points lie on a plane [16].

Both in the case of purely rotating objects (relative to the camera) or in the case of a planar 3-D structure, the motion model between the two views $\mathbf{x}_1 \in \mathbb{P}^2$ and $\mathbf{x}_2 \in \mathbb{P}^2$ is described by a homography matrix $H \in \mathbb{R}^{3 \times 3}$ such that [16]

$$\mathbf{x}_2 \sim H\mathbf{x}_1 = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \mathbf{x}_1. \quad (27)$$

Consider now the case in which we are given a set of image pairs $\{(\mathbf{x}_1^j, \mathbf{x}_2^j)\}_{j=1}^N$ that can be modeled with n *independent* homographies $\{H_i\}_{i=1}^n$ (see Remark 2). Note that the n homographies do *not* necessarily correspond to n different rigid-body motions. This is because it could be the case that one rigidly moving object consists of two or more planes, hence its rigid-body motion will lead to two or more homographies. Therefore, the n homographies can represent anything from 1 up to n rigid-body motions. In either case, it is evident from the form of equation (27) that we cannot take the product of all the equations, as we did with the epipolar constraints, because we have two linearly independent equations per image pair. Nevertheless, we show now that one can still solve the problem by working in the complex domain, as we describe below.

We interpret the second image $\mathbf{x}_2 \in \mathbb{P}^2$ as a point in \mathbb{CP} by considering the first two coordinates in \mathbf{x}_2 as a complex number and appending a one to it. However, we still think of \mathbf{x}_1 as a point in \mathbb{P}^2 . With this interpretation, we can rewrite (27) as

$$\mathbf{x}_2 \sim H\mathbf{x}_1 \doteq \begin{bmatrix} h_{11} + h_{21}\sqrt{-1} & h_{12} + h_{22}\sqrt{-1} & h_{13} + h_{23}\sqrt{-1} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \mathbf{x}_1, \quad (28)$$

where $H \in \mathbb{C}^{2 \times 3}$ now represents a *complex homography*³. Let \mathbf{w}_2 be the vector in \mathbb{CP} perpendicular to \mathbf{x}_2 , i.e. if $\mathbf{x}_2 = (z, 1)$ then $\mathbf{w}_2 = (1, -z)$. Then we can rewrite (28) as the following *complex bilinear constraint*

$$\mathbf{w}_2^T H \mathbf{x}_1 = 0, \quad (29)$$

which we call the *complex homography constraint*. We can therefore interpret the motion segmentation problem as one in which we are given image data $\{\mathbf{x}_1^j \in \mathbb{P}^2\}_{j=1}^N$ and $\{\mathbf{w}_2^j \in \mathbb{CP}\}_{j=1}^N$ generated by a collection of n complex homographies $\{H_i \in \mathbb{C}^{2 \times 3}\}_{i=1}^n$. Then each image pair $(\mathbf{x}_1, \mathbf{w}_2)$ has to satisfy the *multibody homography constraint*

$$\prod_{i=1}^n (\mathbf{w}_2^T H_i \mathbf{x}_1) = \nu_n(\mathbf{w}_2)^T \mathcal{H} \nu_n(\mathbf{x}_1) = 0, \quad (30)$$

regardless of which one of the n complex homographies is associated with the image pair. We call the matrix $\mathcal{H} \in \mathbb{C}^{M_n(2) \times M_n(3)}$ the *multibody homography*. Now, since the multibody homography constraint (30) is linear in the multibody homography \mathcal{H} , we

³ Strictly speaking, we embed each real homography matrix into an affine complex matrix.

can linearly solve for \mathcal{H} from (30) given $N \geq M_n(2)M_n(3) - (M_n(3) + 1)/2 \sim O(n^3)$ image pairs in general position⁴ with at least 4 pairs per moving object.

Given the multibody homography $\mathcal{H} \in \mathbb{C}^{M_n(2) \times M_n(3)}$, the rest of the problem is to recover the individual homographies $\{H_i\}_{i=1}^n$. In the case of fundamental matrices discussed in Section 3.2, the key for solving the problem was the fact that fundamental matrices are of rank 2, hence one can cluster epipolar lines based on the epipoles. In principle, we cannot do the same with real homographies $H_i \in \mathbb{R}^{3 \times 3}$, because in general they are full rank. However, if we work with complex homographies $H_i \in \mathbb{C}^{2 \times 3}$ they automatically have a right null space which we call the *complex epipole* $e_i \in \mathbb{C}^3$. Then, similarly to (26), we can associate a *complex epipolar line*

$$\ell^j \sim \frac{\partial \nu_n(\mathbf{w}_2)^T \mathcal{H} \nu_n(\mathbf{x}_1)}{\partial \mathbf{x}_1} \Big|_{(\mathbf{x}_1, \mathbf{w}_2) = (\mathbf{x}_1^j, \mathbf{w}_2^j)} \in \mathbb{CP}^2 \quad (31)$$

with each image pair $(\mathbf{x}_1^j, \mathbf{w}_2^j)$. Given this set of $N \geq M_n(3) - 1$ complex epipolar lines $\{\ell^j\}_{j=1}^N$, with at least 2 lines per moving object, we can apply Algorithm 1 with $K = 3$ and $\Pi = I$ to estimate the n complex epipoles $\{e_i \in \mathbb{C}^3\}_{i=1}^n$ up to a scale factor, as in equation (23). Therefore, if the n complex epipoles are different, we can cluster the original image measurements by assigning image pair $(\mathbf{x}_1^j, \mathbf{x}_2^j)$ to group i if $i = \arg \min_{\ell=1, \dots, n} |e_\ell^T \ell^j|^2$. Once the image pairs have been clustered, the estimation of each homography, either real or complex, becomes a simple linear problem.

Remark 1 (Direct extraction of homographies from \mathcal{H}). There is yet another way to obtain individual H_i from \mathcal{H} without segmenting the image pairs first. Once the complex epipoles e_i are known, one can compute the following linear combination of the rows of H_i (up to scale) from the derivatives of the multibody homography constraint at e_i

$$\mathbf{w}^T H_i \sim \frac{\partial \nu_n(\mathbf{w})^T \mathcal{H} \nu_n(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{e}_i} \in \mathbb{CP}^2, \quad \forall \mathbf{w} \in \mathbb{C}^2. \quad (32)$$

In particular, if we take $\mathbf{w} = [1, 0]^T$ and $\mathbf{w} = [0, 1]^T$ we obtain the first and second row of H_i (up to scale), respectively. By choosing additional \mathbf{w} 's one obtains more linear combinations from which the rows of H_i can be linearly and uniquely determined.

Remark 2 (Independent homographies). The above solution assumes that the complex epipoles are different (up to a scale factor). We take this assumption as our definition of independent homographies, even though it is more restrictive than saying that the real homographies $H_i \in \mathbb{R}^{3 \times 3}$ are different (up to a scale factor). However, one can show that, under mild conditions, e.g., the third rows of each H_i are different, the null spaces of the complex homographies are indeed different for different real homographies.⁵

⁴ The multibody homography constraint gives two equations per image pair, and there are $(M_n(2) - 1)M_n(3)$ complex entries in \mathcal{H} and $M_n(3)$ real entries (the last row).

⁵ The set of complex homographies that share the same null space is a five-dimensional subset (hence a zero-measure subset) of all real homography matrices. Furthermore, one can complexify any other two rows of H instead of the first two. As long as two homography matrices are different, one of the complexifications will give different complex epipoles.

Remark 3 (One rigid-body motion versus multiple ones). A homography is generally of the form $H = R + T\pi^T$ where π is the plane normal. If the homographies come from different planes (different π) undergoing the same rigid-body motion, the proposed scheme would work just fine since different normal vectors π will cause the complex epipoles to be different. However, if multiple planes with the same normal vector $\pi = [0, 0, 1]^T$ undergo pure translational motions of the form $T_i = [T_{xi}, T_{yi}, T_{zi}]^T$, then all the complex epipoles are equal to $e_i = [\sqrt{-1}, -1, 0]^T$. To avoid this problem, one can complexify the first and third rows of H instead of the first two. The new complex epipoles are $e_i = [T_{xi} + T_{zi}\sqrt{-1}, T_{yi}, -1]^T$, which are different for different translations.

4 Experiments on Real and Synthetic Images

2-D translational. We tested our polynomial differentiation algorithm (PDA) by segmenting 12 frames of a sequence consisting of an aerial view of two robots moving on the ground. The robots are purposely moving slowly, so that it is harder to distinguish the flow from the noise. At each frame, we applied Algorithm 1 with $K = 2$ and $\delta = 0.02$ to the optical flow⁶ of all $N = 240 \times 352$ pixels in the image and segmented the image measurements into $n = 3$ translational motion models. The leftmost column of Figure 1 displays the x and y coordinates of the optical flow for frames 4 and 10, showing that it is not so simple to distinguish the three clusters from the raw data. The remaining columns of Figure 1 show the segmentation of the image pixels. The motion of the two robots and that of the background are correctly segmented. We also applied Algorithm 1 to the optical flow of the flower garden sequence. Figure 2 shows the optical flow of one frame and the segmentation of the pixels into three groups: the tree, the grass, and the background. Notice that the boundaries of the tree can be assigned to any group, and in this case they are grouped with the grass.

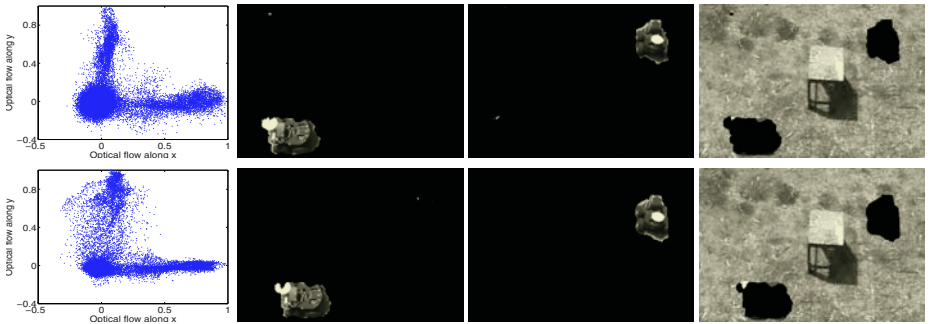


Fig. 1. Segmenting the optical flow of the two-robot sequence by clustering lines in \mathbb{C}^2

⁶ We compute optical flow using Black's code at <http://www.cs.brown.edu/people/black/ignc.html>.

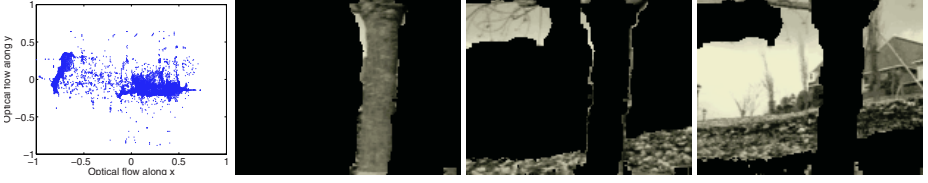


Fig. 2. Segmenting the optical flow of the flower-garden sequence by clustering lines in \mathbb{C}^2

3-D translational motions. Figure 3(a) shows the first frame of a 320×240 video sequence containing a truck and a car undergoing two 3-D translational motions. We applied Algorithm 1 with $K = 3$, $\Pi = I$ and $\delta = 0.02$ to the (real) epipolar lines obtained from a total of $N = 92$ features, 44 in the truck and 48 in the car. The algorithm obtained a perfect segmentation of the features, as shown in Figure 3(b), and estimated the epipoles with an error of 5.9° for the truck and 1.7° for the car. We also tested the performance of PDA on synthetic data corrupted with zero-mean Gaussian noise with s.t.d. between 0 and 1 pixels for an image size of 500×500 pixels. For comparison purposes, we also implemented the polynomial factorization algorithm (PFA) of [12] and a variation of the Expectation Maximization algorithm (EM) for clustering hyperplanes in \mathbb{R}^3 . Figures 3(c) and (d) show the performance of all the algorithms as a function of the level of noise for $n = 2$ moving objects. The performance measures are the mean error between the estimated and the true epipoles (in degrees), and the mean percentage of correctly segmented features using 1000 trials for each level of noise. Notice that

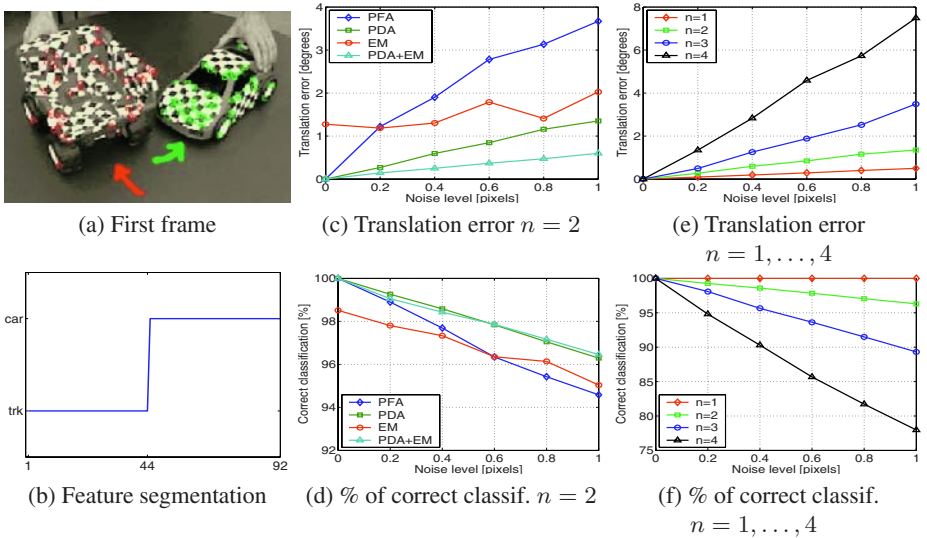


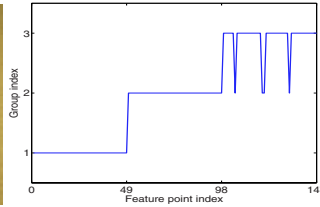
Fig. 3. Segmenting 3-D translational motions by clustering planes in \mathbb{R}^3 . Left: segmenting a real sequence with 2 moving objects. Center: comparing our algorithm with PFA and EM as a function of noise in the image features. Right: performance of PFA as a function of the number of motions

PDA gives an error of less than 1.3° and a classification performance of over 96%. Thus our algorithm PDA gives approximately 1/3 the error of PFA, and improves the classification performance by about 2%. Notice also that EM with the normal vectors initialized at random (EM) yields a nonzero error in the noise free case, because it frequently converges to a local minimum. In fact, our algorithm PDA outperforms EM. However, if we use PDA to initialize EM (PDA+EM), the performance of both EM and PDA improves, showing that our algorithm can be effectively used to initialize iterative approaches to motion segmentation. Furthermore, the number of iterations of PDA+EM is approximately 50% with respect to EM randomly initialized, hence there is also a gain in computing time. Figures 3(e) and (f) show the performance of PDA as a function of the number of moving objects for different levels of noise. As expected, the performance deteriorates with the number of moving objects, though the translation error is still below 8° and the percentage of correct classification is over 78%.

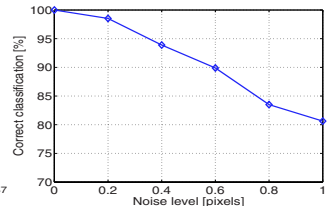
3-D homographies. Figure 4(a) shows the first frame of a 2048×1536 video sequence with two moving objects: a cube and a checkerboard. Notice that although there are only *two* rigid motions, the scene contains *three* different homographies, each one associated with each one of the visible planar structures. Furthermore, notice that the top side of the cube and the checkerboard have approximately the same normals. We manually tracked a total of $N = 147$ features: 98 in the cube (49 in each of the two visible sides) and 49 in the checkerboard. We applied our algorithm in Section 3.3 with $\Pi = I$ and $\delta = 0.02$ to segment the image data and obtained a 97% of correct classification, as shown in Figure 4(b). We then added zero-mean Gaussian noise with standard deviation between 0 and 1 pixels to the features, after rectifying the features in the second view in order to simulate the noise free case. Figure 4(c) shows the mean percentage of correct classification for 1000 trials per level of noise. The percentage of correct classification of our algorithm is between 80% and 100%, which gives a very good initial estimate for any of the existing iterative/optimization/EM based motion segmentation schemes.



(a) First frame



(b) Feature segmentation



(c) % of correct classification

Fig. 4. Segmenting 3-D homographies by clustering complex bilinear forms in $\mathbb{C}^{2 \times 3}$

5 Conclusions

We have presented a unified algebraic approach to 2-D and 3-D motion segmentation from feature correspondences or optical flow. Contrary to extant methods, our approach does not iterate between feature segmentation and motion estimation. Instead, it computes a single multibody motion model that is satisfied by all the image measurements and then extracts the original motion models from the derivatives of the multibody one. Various experiments showed that our algorithm not only outperforms existing algebraic methods with much limited applicability, but also provides a good initialization for iterative techniques, such as EM, which are strongly dependent on correct initialization.

References

1. Darrel, T., Pentland, A.: Robust estimation of a multi-layered motion representation. In: IEEE Workshop on Visual Motion. (1991) 173–178
2. Jepson, A., Black, M.: Mixture models for optical flow computation. In: CVPR. (1993) 760–761
3. Ayer, S., Sawhney, H.: Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In: ICCV. (1995) 777–785
4. Weiss, Y.: Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In: CVPR. (1997) 520–526
5. Shi, J., Malik, J.: Motion segmentation and tracking using normalized cuts. In: ICCV. (1998) 1154–1160
6. Torr, P., Szeliski, R., Anandan, P.: An integrated Bayesian approach to layer extraction from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23** (2001) 297–303
7. Wang, J., Adelson, E.: Layered representation for motion analysis. In: CVPR. (1993) 361–366
8. Feng, X., Perona, P.: Scene segmentation from 3D motion. In: CVPR. (1998) 225–231
9. Vidal, R., Sastry, S.: Segmentation of dynamic scenes from image intensities. In: IEEE Workshop on Vision and Motion Computing. (2002) 44–49
10. Costeira, J., Kanade, T.: Multi-body factorization methods for motion analysis. In: ICCV. (1995) 1071–1076
11. Kanatani, K.: Motion segmentation by subspace separation and model selection. In: ICCV. (2001) 586–591
12. Vidal, R., Ma, Y., Sastry, S.: Generalized principal component analysis (GPCA). In: CVPR. (2003) 621–628
13. Wolf, L., Shashua, A.: Two-body segmentation from two perspective views. In: CVPR. (2001) 263–270
14. Vidal, R., Ma, Y., Soatto, S., Sastry, S.: Two-view multibody structure from motion. To appear in *International Journal of Computer Vision* (2004)
15. Vidal, R., Ma, Y., J. Piazzi: A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing polynomials. In: CVPR. (2004)
16. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge (2000)

Enhancing Particle Filters Using Local Likelihood Sampling

Péter Torma¹ and Csaba Szepesvári²

¹ Tateyama Ltd., Zenta u. 1.
1111 Budapest, Hungary
tyus@tateyama.hu

² Computer Automation Institute of the Hungarian Academy of Sciences,
Kende u. 13-17, 1111, Budapest, Hungary
Csaba.Szepesvari@sztaki.hu; <http://www.sztaki.hu/szepesvari>

Abstract. Particle filters provide a means to track the state of an object even when the dynamics and the observations are non-linear/non-Gaussian. However, they can be very inefficient when the observation noise is low as compared to the system noise, as it is often the case in visual tracking applications. In this paper we propose a new two-stage sampling procedure to boost the performance of particle filters under this condition. We provide conditions under which the new procedure is proven to reduce the variance of the weights. Synthetic and real-world visual tracking experiments are used to confirm the validity of the theoretical analysis.

1 Introduction

In this paper we consider particle filters in the special case when the observation noise is low as compared to the noise in the system’s dynamics (for brevity we call the latter noise the ‘system noise’). This is a typical situation in computer vision where the discrimination power of the object model is typically high. Such models may e.g. use shape, contour, colour, intensity information or a combination of these and give rise to a highly peaked, low entropy observation likelihood function. Highly discriminative observation likelihoods are very desirable as they results in highly peaked posteriors and hence, in theory, the position of the object can then be estimated with high precision.

In practice the posterior cannot be obtained in a closed form, except in a few special cases. Thus in general one must revert to some approximate method to estimate the posterior. Particle filters represent a rich class of such approximate methods. They represent the posterior using a weighted particle set living in the state space of the process. Upon the receipt of a new observation the generic particle filter algorithm updates the position of the particles and recomputes the weights so that the new weighted sample becomes a good representation of the new posterior that takes into account the new observation, as well. The position of the particles are typically updated independently of each other by drawing them from a user-chosen proposal distribution. If the new particles are not in the

close vicinity of the modes of the likelihood function and the likelihood function is highly peaked then the representation of the posterior will degrade very fast. The peakier the observation likelihood function the more important is to sample the particles such that they will be close to the modes.

This paper introduces a method that draws the new positions of the particles using a two-stage sampling process that depends on the likelihood function and hence one expects the new method to have superior performance than that of those algorithms that do not use the likelihood function. At the core of the new filter is sampling method that is applicable when the density to sample from has a product form, with one of the terms being highly peaked, whilst the other having heavy tails. Under this conditions the new sampling method represents an alternative to importance sampling. The new method works by first drawing a particle from the broader density. In the second step this particle is perturbed such that on average it moves closer to one of the modes of the peaky term. Weights are calculated so that unbiasedness of the new sample is guaranteed. We compare the expected performance of the proposed scheme by means of a theoretical analysis to that of the basic importance sampling scheme and derive conditions under which the new scheme can be expected to perform better. The comparison is extended to the particle filtering setting, as well. The theoretical findings are confirmed in some computer experiments. In particular, the method is applied to the tracking of Japanese license plates where the new algorithm is shown to improve performance substantially both in terms of tracking accuracy and speed.

1.1 Related Work

The efficiency problem associated with low observation noise is well known in the literature and hence many approaches exist to resolving it. Among the many methods the Auxiliary variable Sampling Importance Resampling (ASIR) filter introduced by Pitt and Shephard [1] is one of the closest to our algorithm. ASIR approximates the proposal density using a mixture of the form $\sum_{k=1}^N \beta_k \gamma_k(\cdot)$ where weight β_k approximates the normalized likelihood of the new observation (Y_t) assuming that the state of the process at the $t - 1$ th time step is $X_{t-1}^{(k)}$. The function $\gamma_k(\cdot)$ approximates the density $p(x_t | X_{t-1}^{(k)}, Y_1, \dots, Y_t)$.¹ The biggest obstacle in applying ASIR is to obtain a good approximation of the conditional likelihood of the new observation given $X_{t-1}^{(k)}$, since this involves the evaluation of a potentially high dimensional integral and the efficiency of ASIR ultimately depends on the quality of this approximation. It turns out that except for some special cases it is not easy to come up with a sampling scheme that could sample from $p(x_t | X_{t-1}^{(k)}, Y_1, \dots, Y_t) \propto p(Y_t | x_t) p(x_t | X_{t-1}^{(k)})$ in an efficient manner. This issue is the problem we are addressing in this paper.

Another recent proposal is called ‘likelihood sampling’ (see e.g. [2]). In this approach it is the likelihood function $p(Y_t | \cdot)$ that is used to generate the samples

¹ Here and in what follows random variables are denoted by big capitals.

from and the prediction density is used to calculate the weights. The success of this method depends on the amount of ‘state aliasing’ that comes from the limitations of the observation model. For multi-modal observation likelihoods a large number of particles can be generated that will have low weights when the posterior already concentrates on a small part of the state space. Our method overcomes this problem by first sampling from the prediction density. As a result, our method can suffer from inefficiency if the estimates posterior is considerably off from the target so we expect our method being competitive only when this is not the case.

Perhaps the most relevant to this work is the LS-N-IPS algorithm introduced in [3]. In LS-N-IPS the prediction density is used to derive the new particle set which is then locally modified by climbing the observation likelihood. Hence this algorithm introduces some bias and also needs an efficient method to climb the observation likelihood. The method proposed here resolves the inefficiency problem without introducing any additional bias or requiring a hill-climbing method.

2 Notation

The following notations will be used: for an integrable function f , $I(f)$ will denote the integral of f with respect to the Lebesgue measure.² \mathbb{R}^d denotes the d -dimensional Euclidean space. L^p ($0 < p \leq +\infty$) denotes the set of functions with finite p -norm. The p -norm of a function is denoted by $\|f\|_p$. For a function $f \in L^s(\mathbb{R}^d)$, $s \in \{1, 2\}$, \hat{f} denotes its Fourier transform: $\hat{f}(\omega) = \int e^{-i\omega^T x} f(x) dx$. The inner product defined over $L^2(\mathbb{R}^d)$ is defined by $\langle f, g \rangle = \int f(x) \bar{g}(x) dx$, where \bar{a} denotes the complex conjugate of a . Convolution is denoted by $*$: $(f * g)(x) = \int f(y) g(x - y) dy$. Expectation is denoted by E and variance by Var , as usual.

3 Random Representation of Functional Products

Our main interest is to generate random samples that can be used to represent products with two terms f, p , where f is an integrable function (f plays the role of the observation likelihood) and p is a density (the prediction density). We begin with the definition of what we mean by a properly weighted set w.r.t. f and p . This definition is a slightly modified version of the definition given in [4]:

Definition 1 *A random variable X is said to be properly weighted by the function w with respect to the density p and the integrable function f if for any integrable function h , $E[h(X)w(X)] = \int h(x)f(x)p(x) dx$. Also, in this case $(X, w(X))$ is said to form a properly weighted pair with respect to f, p .*

² The underlying domain of the functions is not important at this point. It could be any Polish set, e.g. an Euclidean space.

A set of random draws and weight functions $\{(X_j, w_j)\}_{j=1, \dots, N}$ is said to be properly weighted with respect to p and f if all components of the set are properly weighted with respect to p and f .

It should be clear that if $\{(X_j, w_j)\}_{j=1, \dots, N}$ is a properly weighted set with respect to p and f then, by the law of large numbers, the sample averages $J_N(h, w) = (1/N) \sum_{j=1}^N h(X_j)w(X_j)$ will converge to $I(hfp)$ under fairly mild conditions. Hence, in this sense, a properly weighted set with respect to f and p can thought of as representing the product density $f(\cdot)p(\cdot)$.

Let us now consider two constructions for properly weighted sets. It is clear from the definition that it is sufficient to deal with the case of a single random variable.

The most obvious way to obtain a properly weighted pair is to sample X from p and define $w = f$. Then, trivially $E[h(X)w(X)] = E[h(X)f(X)] = I(hfp)$. We shall call this the *canonical* or *basic sampling scheme*.

A central question of Monte-Carlo sampling is how to obtain a properly weighted set such that the variance of the estimate of $I(hfp)$ provided by the sample average $J_N(h, w)$ is minimized. Actually, we are more interested in studying the weight-normalized averages $I_N(h, w) = \frac{\sum_{j=1}^N h(X_j)w(X_j)}{\sum_{j=1}^N w(X_j)}$ that converge to the normalized value $I(hfp)/I(fp)$ as $N \rightarrow \infty$ with probability one, under a broad range of conditions on (X_j, w_j) . Obviously, the optimal sampling construction depends on h . Since we are interested in the case when h is not fixed, it is sensible to use the “rule of thumb” presented in Liu [5] (based on [6]) to measure the efficiency of a sampling construction by a quantity inversely proportional to the variance of $w(X)$, where w is such that $E[w(X)] = 1$. By straightforward calculations one can show that Liu’s measure still applies to our case, because $E[w(X)] = I(fp) \equiv \text{const}$, independently of the choice of X and w . We make this rule as our starting point and will compare different sampling schemes by the variance of the weights. Now, since for any properly weighted pair (X, w) , $E[w(X)] = I(fp)$ and $\text{Var}[w(X)] = E[w^2(X)] - E[w(X)]^2$ we find that minimizing $\text{Var}[w(X)]$ is equivalent to minimizing $E[w^2(X)]$. Note that if X is drawn from p and w is set to be equal to f , then $E[w^2(X)] = I(f^2p)$.

The sampling scheme we propose works by locally perturbing the samples drawn from p to move them closer to the modes of f . Let $g \in L^1$ be a compactly supported function with $I(g) = 1$.

Locally Perturbed Sampling Procedure

1. Draw N independent samples X_1, \dots, X_N from p .
2. For each $1 \leq j \leq N$, draw samples Z_j from $\frac{f(z)g(X_j - z)}{(f * g)(X_j)} dz$.
3. Calculate the weights $w_j = (f * g)(X_j)p(Z_j)/p(X_j)$ and output $\{(Z_j, w_j)\}$.

The algorithm first draws samples from p , just like the canonical one. In the second step, the samples are ‘moved’ towards the modes of f , but stay in the close vicinity of the drawn samples thanks to the compact support of g .³ Hence

³ A slight variant can be obtained by employing a two-variable kernel function $G(x, z)$ in place of $g(x - z)$. Then in the algorithm $(f * g)(X_j)$ is replaced by $\int f(z)G(x, z)dz$.

the name of the procedure. In the next step, weights are calculated so that (Z_j, w_j) becomes a properly weighted pair for f, p .

In the above algorithm the function g could be a truncated Gaussian, or the characteristic function of some convex set. In one extreme but practical case g equals the weighted sum of translated Dirac-delta functions: $g_m(x) = \sum_{k=1}^m v_k \delta(x - t_k)$. We shall call such a function a Dirac-comb. When g is a Dirac-comb, the random sample Z_j is drawn from the weighted discrete distribution $\{(X_j + t_l, p_l)\}$, where $p_l \propto f(X_j + t_l)$.

The following proposition shows that the proposed sampling scheme results in unbiased samples.⁴

Proposition 1 *Assume that $g \in L^1$ is a compactly supported function satisfying $g \geq 0$ and $I(g) = 1$. Let f be a bounded, integrable function and let p be a density. Then, the above sampling procedure yields properly weighted pairs (Z_j, w_j) with respect to f, p .*

The efficiency of the scheme will obviously depend on the correlation of f and p : if the modes of p were far away from the modes of f then the scheme will be inefficient. Another source of inefficiency is when the support of g is too small to move the samples to the vicinity of the modes of f or when it is too large. In the limit when the support of g grows to \mathcal{X} the scheme reduces to likelihood sampling. The following proposition provides the basic ground for the analysis of the efficiency of this scheme.

Proposition 2 *Assume that $g \in L^1$ is an even, compactly supported function satisfying $g \geq 0$ and $I(g) = 1$ and let f be a bounded, integrable, nonnegative function and let p be a density. Define the operator $A : L^1 \rightarrow L^\infty$ by*

$$(Ah)(u) = \begin{cases} \int h(t)p(t)g(t-u) \left(\frac{p(t)}{p(u)} - 1 \right) dt, & \text{when } p(u) > 0; \\ 0, & \text{otherwise.} \end{cases}$$

*Assume that for some $s \in [1, \infty]$, $\epsilon = \sup_{h \in L^1, h \geq 0} \sup_u (Ah)(u) / \|h\|_s < +\infty$. Let (Z, w) be a random sample as defined in the above algorithm. Then $E[w^2] \leq \langle f * g, fp * g \rangle + \epsilon I(f) \|f\|_s$.*

From this proposition it follows immediately that the proposed scheme is more efficient than the canonical algorithm whenever $\epsilon I(f) \|f\|_s \leq \langle f, fp \rangle - \langle f * g, (fp) * g \rangle$. Clearly, this formula agrees well with our earlier intuition: the right hand side is maximized, when the cross-correlation of f and fp is high and the cross-correlation of $f * g$ and $(fp) * g$ is small. In some cases convolution with g can thought of as a low-pass filtering operation (e.g. think about when g is the characteristic function of the unit interval) and hence g cuts some of the high frequency of f and fp . As a result, the cross-correlation of $f * g$ and $(fp) * g$ can be expected to be smaller than the cross-correlation of f and fp .

⁴ Here and in what follows the proofs are omitted due to a lack of space. An extended version of the paper available on the website of the authors contains all the missing proofs. The proof of this proposition uses Fubini's theorem and $I(f * g) = I(f)$.

It remains to see, however, whether A is bounded and ϵ is well defined. For this define $p_d(x) = \inf_{\|y\| < d} p(x + y)$. Then, one can show that for $s = 1$, $\epsilon \leq \|p\|_\infty \|p/p_d\|_\infty < +\infty$. Of course, this bound is not particularly tight and much tighter bounds can be derived in special cases. For example when g is equal to the Dirac-comb defined earlier and if $\max_k \|t_k\| \leq d$ then $(Af)(u) \leq \|f\|_\infty \|p\|_\infty \|p/p_d\|_\infty$. Therefore, in this case Proposition 2 holds with $s = +\infty$. By means of some convexity arguments one may also derive bounds for mixture densities. This can be useful to derive sharper bounds on ϵ .

4 Particle Filters Enhanced by Local Likelihood Sampling

Let us now consider the problem of filtering a non-linear system of the form $X_t = a(X_{t-1}) + W_t$, $Y_t = b(X_t) + V_t$, where $X_t \in \mathcal{X}$ is the state of the system⁵ at time t and $Y_t \in \mathbb{R}^p$ is the observation at time t . We assume that $X_0 \sim p_0$. Here $W_1, V_1, W_2, V_2, \dots$ are independent, W_1, W_2, \dots are identically distributed, just like V_1, V_2, \dots . For the sake of simplicity, we further assume that the densities $K(x|x') = p(X_t = x | X_{t-1} = x')$ and $f(y|x) = p(Y_t = y | X_t = x)$ exist. The problem we consider is the estimation of the posterior $p(X_t | Y_{1:t})$, where $Y_{1:t}$ denotes the sequence of past observations: $Y_{1:t} = (Y_1, \dots, Y_t)$.

Particle filters approximate the posterior by a random measure $\pi_t(x) = (\sum_{k=1}^N w_t^{(k)} \delta(x - X_t^{(k)})) / \sum_{k=1}^N w_t^{(k)}$, where $X_t^{(k)}$ are called the particles, and $w_t^{(k)}$ is the weight of the i th particle. $X_t^{(k)}, w_t^{(k)}$ are random quantities and depend on the sequence of past observation $Y_{1:t}$. The best known particle filter is probably the SIR⁶ filter [7], also known as CONDENSATION [8]:

SIR Filter

1. Draw N independent samples $X_0^{(1)}, \dots, X_0^{(N)}$ from p_0 .
2. Repeat for $t = 1, 2, \dots$:
 - a) Draw $\hat{X}_t^{(k)} \sim q(x_t | X_{t-1}^{(k)}, Y_t)$, $k = 1, \dots, N$ independently of each other.
 - b) Calculate the weights

$$w_t^{(k)} = \left(f(Y_t | \hat{X}_t^{(k)}) K(\hat{X}_t^{(k)} | X_{t-1}^{(k)}) \right) / q(\hat{X}_t^{(k)} | X_{t-1}^{(k)}, Y_t).$$
 - c) Draw a sequence of independent indexes j_1, \dots, j_N such that $p(j_l = k) \propto w_t^{(k)}$ and set $X_t^{(k)} = \hat{X}_t^{(j_k)}$. The corresponding weights are set to $1/N$.

SIR uses importance sampling to sample from $\frac{f(Y_t | \cdot) K(\cdot | X_{t-1}^{(k)})}{p(Y_t | Y_{1:t})}$, hence its efficiency will depend on how well the proposal density q matches the shape of this function. A particularly popular choice for the proposal q is the *prediction density* K : $q(x_t | X_{t-1}^{(k)}, Y_t) = K(x_t | X_{t-1}^{(k)})$. In this case drawing from q is equivalent to simulating the dynamics of the system for a single time-step starting from $X_{t-1}^{(k)}$. This is typically simple to implement, hence the popularity of this

⁵ Typically we will have $\mathcal{X} = \mathbb{R}^d$.

⁶ Sampling Importance Resampling

choice. Also, in this case the weights become particularly simple to compute: $w_t^{(k)} = f(Y_t|X_t^{(k)})$. We shall call the SIR algorithm with this choice the “basic” or “canonical” SIR algorithm.

Following the “rule of thumb” discussed in the previous section we shall measure the efficiency of sampling at time step t by $\text{Var}[w_t^{(k)}|X_{t-1}^{(k)}, Y_{1:t}]$.

Our proposed new particle filter uses the method of the previous section to boost the performance of this algorithm in the case of low observation noise:

Local Likelihood Sampling SIR (LLS-SIR)

1. Draw N independent samples $X_0^{(1)}, \dots, X_0^{(N)}$ from p_0 .
2. Repeat for $t = 1, 2, \dots$:
 - a) Draw $\hat{X}_t^{(k)} \sim K(\cdot|X_{t-1}^{(k)})$, $k = 1, \dots, N$, independently of each other.
 - b) Draw $Z_t^{(k)} \sim \frac{1}{\alpha_t^{(k)}} f(Y_t|\cdot) g(\hat{X}_t^{(k)} - \cdot)$,⁷ independently of each other, where $\alpha_t^{(k)} = \int f(Y_t|x) g(\hat{X}_t^{(k)} - x) dx$.
 - c) Calculate the weights $w_t^{(k)} = \alpha_t^{(k)} K(Z_t^{(k)}|X_{t-1}^{(k)}) / K(\hat{X}_t^{(k)}|X_{t-1}^{(k)})$.
 - d) Resample from $\{(Z_t^{(k)}, p_t^k)\}$ with $p_t^k \propto w_t^{(k)}$ just like it was done in SIR to get the particles $X_t^{(k)}$. Set the weights uniformly to $1/N$.

The algorithm is identical to SIR except that the new particle positions are determined using the two-stage sampling procedure introduced in the previous section.

The following proposition shows the 1-step unbiasedness of the algorithm:

Proposition 3 *Assume that g is a non-negative, compactly supported, integrable function satisfying $I(g) = 1$. Then LLS-SIR does not introduce any more bias than the SIR algorithm in the sense that for any integrable function h one has*

$$E[w_t^{(k)} h(Z_t^{(k)}) | Y_{1:t}] = E[h(X_t) | Y_{1:t}] p(Y_t | Y_{1:t-1}).$$

As a consequence of this proposition, convergence results analogous to those that are known for the SIR algorithm can be derived.

Now, we compare the efficiency of the proposed algorithm with that of the basic SIR algorithm. We shall focus on the case when g is a Dirac-comb since this choice allows one to implement the filter for continuous state spaces which is the case that we are particularly interested in.

5 Variance Analysis: The Case of the Dirac-Comb

In this section for the sake of simplicity we consider one-dimensional systems only. Note that these results extend to multi-dimensional systems without any

⁷ Here g is a compactly supported, integrable, nonnegative function satisfying $I(g) = 1$ as before.

problems.⁸ We shall consider the choice $g_m(x) = \frac{1}{m} \sum_{l=-(m-1)/2}^{(m-1)/2} \delta(x - l\lambda)$, where $m, \lambda > 0$.

The following theorem expresses the difference between the appropriately normalized variance of basic SIR and that of LLS-SIR. The normalization is intended to compensate for the m -times larger number of likelihood calculations required by LLS-SIR. The estimate developed here shows that LLS-SIR is more advantageous than SIR when started from the same particle set.

Theorem 1 *Assume that both the basic SIR and the proposed Local Likelihood Sampling algorithm each draw $X_t^{(k)}$ from $K(\cdot | X_{t-1}^{(k)})$, where $X_{t-1}^{(k)}$ is a common sample from a density $\hat{p}(\cdot) \approx p(\cdot | Y_{1:t-1})$. Let $w_t^{(k)}(\text{SIR})$ and $w_t^{(k)}(P)$ denote the (unnormalized) weights of the basic SIR and the Local Likelihood Sampling algorithm, respectively. Let $\Delta = \frac{1}{Nm} \text{Var}[w_t^{(k)}(\text{SIR}) | X_{t-1}^{(k)}, Y_{1:t}] - \frac{1}{N} \text{Var}[w_t^{(k)}(P) | X_{t-1}^{(k)}, Y_{1:t}]$. Let $\epsilon, s > 0$ be defined as in Proposition 2, where in the definition of the operator A one uses $p(\cdot) = K(\cdot | X_{t-1}^{(k)})$. Let $f(\cdot) = f(Y_t | \cdot)$. Then $N\Delta \geq (\langle f, fp \rangle - \langle f * g, (fp) * g \rangle) - (\frac{m-1}{m} \text{Var}_p[f] + \epsilon I(f) \|f\|_s)$. Hence, the proposed sampling scheme is more efficient than the one used by SIR provided that*

$$\langle f, p \rangle^2 \geq \langle f * g, (fp) * g \rangle + \epsilon I(f) \|f\|_s. \quad (1)$$

Let us now specialize (1) to the case when g equals to the equidistant Dirac-comb defined earlier. By using harmonic analysis arguments, one gets $m/2 \langle f * g_m, fp * g_m \rangle \rightarrow 1/(2\pi) I(f) I(fp)$, as $m \rightarrow \infty$. Hence, $\langle f * g_m, fp * g_m \rangle \sim \frac{1}{m\pi} I(f) I(fp)$. Hence, condition 1 can be approximated by $\langle f, p \rangle^2 \geq \frac{1}{m\pi} I(f) I(fp) + \epsilon_m I(f) \|f\|_\infty$. Here, we have used ϵ_m instead of ϵ in order to emphasize the dependency of ϵ on m . In general ϵ_m may (and often will) diverge to infinity as $m \rightarrow \infty$.⁹ As a result we get a tradeoff as a function of ϵ_m . In general one expects that when condition (1) is satisfied then it will be satisfied for an interval of values of m .

6 Experiments and Results

6.1 Simulation

We have simulated the system $x_t = x_{t-1}/2 + 25x_{t-1}/(1 + x_t^2) + 8\cos(1.2t) + W_t$, $y_t = |x_t|/20 + V_t$, where $W_t \sim N(0, 10)$ and $V_t \sim N(0, 2)$.¹⁰ LLS-SIR was implemented by using a Gaussian kernel $G_\sigma(x, z) \propto \exp(-(\text{sgn}(x)x^2/20 - z^2/20)^2/(2\sigma^2))$. In this case $f(Y_t | z)G(X_t, z)$ becomes a Gaussian in z^2 and hence one can use importance sampling to sample Z from the corresponding density.

⁸ Note, however, that for high-dimensional state spaces more efficient schemes are needed. One such scheme is given in Section 6.

⁹ Note that when the state space \mathcal{X} is compact then ϵ_m stays bounded.

¹⁰ This system is a slight variant of one system that has been used earlier in a number of papers, in particular in the observation function we used $|x_t|$ instead of x_t^2 .

The system was simulated for $T = 60$ time steps and we have measured the performance in terms of the average RMSE of $|x_t|$. The number of particles was $N = 50$. The RMSE results obtained are 2.12 and 0.94 for SIR and LLS-SIR, respectively, whilst LLS-SIR took 1.1 times more time. Thus, in this case LLS-SIR is slightly more expensive than SIR, but this may well pay off in its increased accuracy.

6.2 Visual Tracking

The proposed algorithm was tested and compared with basic SIR on the problem of tracking Japanese license plates. In these experiments, the outline of a license plate was taken as a parallelogram with two vertical lines.¹¹

Japanese license plates enjoy a very specific geometrical structure, see Figure 1. This gives the basic idea of the observation model. The observation model is scale-free and the likelihood is expressed as the product of the likelihoods of the parts of the license plate where the parts are looked for at the precise locations as dictated by the geometry of the license plates: For each designated area of a candidate plate we compute the likelihood of the observed pattern assuming that the area is of the “right type”. The calculus of these likelihoods is implemented in an ad hoc manner using simple image processing operations that rely on measuring frequency content in the spatial domain. Based on a larger sample of images we have found that the likelihood is sufficiently specific to these kind of license plates. An example image is included in Figure 3.

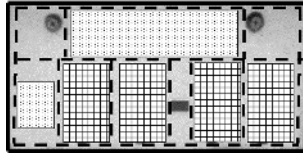


Fig. 1. The model of a Japanese license plate. Checked means “thick line” area, dotted means “thin line” area, dashed line means “clear” area, solid line means “edge”

The object dynamics is a mixture of an initial distribution p_0 and a simple AR(2) product-process: in each time step with a constant probability we assume that the license plate reappears at a random position unrelated to the previous position. The probability of this event is set to a small value (0.1 in the experiments described here). Separate AR(2) models were used to evolve the position, scale, orientation and aspect ratio of the plate, each one independently of the others. The parameters of the AR(2) model were tuned by hand by conducting short preliminary experiments.

¹¹ Thus the configuration of a license plate on the image can be defined with 5 parameters (assuming a fixed aspect ratio).

LLS was implemented only for the position of the plates. Denoting positions by (x_1, x_2) and by $p(x_1, x_2)$ the corresponding weights used in the LLS step, we used $p(x_1, x_2) = p(x_1|x_2)p(x_2)$ in the sampling step as follows. We sample first x_2 from $p(x_2)$, and then x_1 from $p(\cdot|x_2)$. The distribution $p(x_2) = \sum_{x_1} p(x_2, x_1)$ was approximated by making the corresponding observation likelihoods insensitive to the precise horizontal location. The search length was half of the size of the predicted plate size, in both directions.

Results. The performance of the local sampling algorithm with $N = 100$ samples was compared to the performance of the basic SIR filter using $N = 750$ particles. The number of particles for LLS-SIR was preestimated so that we expected the two algorithms to have roughly equivalent running times. It turned out that both algorithms were capable of running faster than real-time. In particular, on our 1.7GHz Intel test machine we have measured a processing speed of approximately 48 frames per seconds for the basic SIR algorithm, whilst for LLS-SIR the measured processing speed was approximately 65 frames per seconds, i.e., we have slightly overestimated the resource requirements of LLS-SIR.

Performance evaluation was done as follows: We selected a test video sequence that consisted of 298 frames. In each time step particle locations were averaged to get the final guess of the license plate position. This position was compared to the “ground truth” obtained by running basic SIR for the test video sequence with $N = 10,000$ of particles and then correcting the results manually. Some frames of this sequence are shown in Figure 2. To be able to judge the difficulty of the tracking task Figure 3 shows the observation likelihood function of a selected frame. On this image the intensity of a pixel is proportional to the logarithm

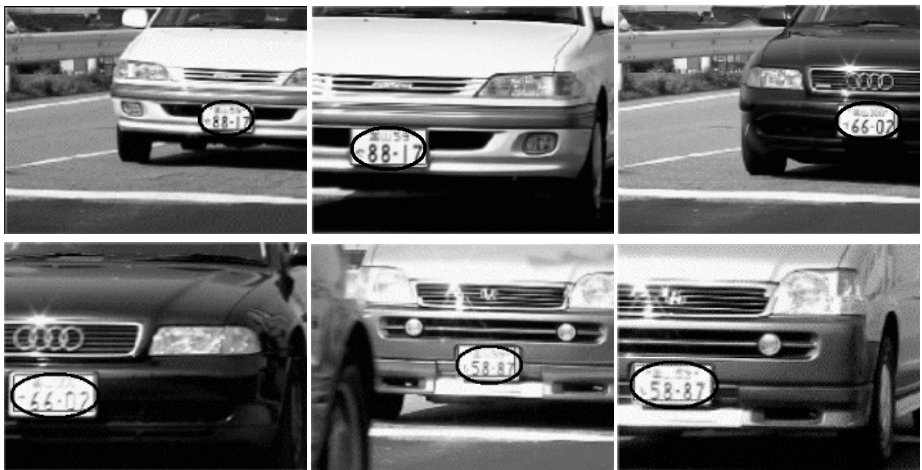


Fig. 2. Sample images of the test video sequence. The video sequence is recorded by a commercial NTSC camera. The frame indexes of the images are 9,29,82,105,117 and 125. The plate positions predicted by LLS-SIR are projected back on the image

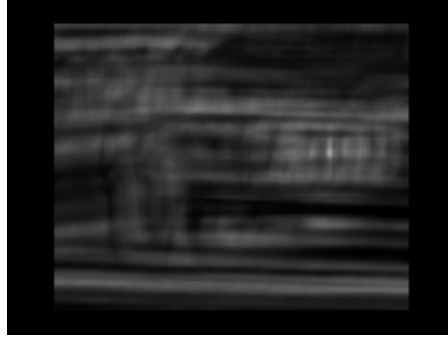


Fig. 3. Log-likelihood of a selected frame of the test video sequence. Note that pixel intensities are taken for the maximum of the logarithm of the observation likelihood where scale is kept free. For more information see the text

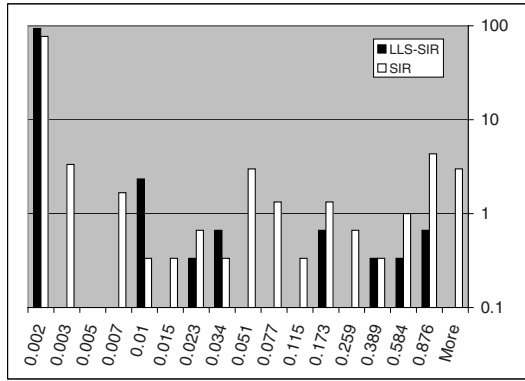


Fig. 4. Histogram of the probability of not tracking the object. Note the log-scales of the axis

of the maximum of the observation likelihood where the maximum is taken for plate configurations with the center of gravity of plates matching the pixel’s position, the orientation matching the best orientation, but keeping the scale of the plates free.

Define the distance of two license plate configurations as the sum of distances of their corresponding vertex points. If this distance is larger than one third of the license plate height then the license plate is considered to be “lost”. The probability of this event was estimated for each frame by means of running $n = 100$ Monte-Carlo experiments. Figure 4 shows the histogram of the probabilities of object loss on the test sequence. Note the log-scale of both axis. The percentage of frames when LLS-SIR tracks the plates (i.e it never loses the plate in any of the experiments) is over 94%. The corresponding number is 77% for SIR. Tracking error was measured for those frames when the object was tracked by the respective algorithms. The median tracking error is 1.23 pixels and 5.36

pixels for LLS-SIR and SIR, respectively. The corresponding means are 3.62 and 5.29, the standard deviations are 4.88 and 5.91. Thus, we conclude that in this case LLS-SIR is more efficient than the basic SIR algorithm both in terms of execution speed and tracking performance.

7 Conclusions

We have proposed a new algorithm, LLS-SIR to enhance particle filters in the low observation noise limit. The algorithm is a modification of the standard particle filter algorithm whereas after the prediction step the position of the particles are randomly resampled from the localized observation density. Theoretical analysis revealed that the scheme does not introduce any bias as compared to the basic SIR algorithm. It was also shown that the new algorithm achieves a higher effective sample size than the basic one when the observations are reliable. This results in a better tracking performance, as it was illustrated on a synthetic and a real world tracking problem.

Further work shall include a more thorough evaluation of the proposed algorithm and more comparisons with competing algorithms. On the theoretical side, extending previous uniform convergence results to the new algorithm looks like an interesting challenge. Another important avenue of research is to extend the results of Section 5 so that one can compare the long term behaviour of the various algorithms. Derivation of lower bounds on the tracking accuracy could be another important next step, too.

References

1. Pitt, M.K., Shephard, N.: Filtering via simulation: Auxiliary particle filter. *Journal of the American Statistical Association* **94** (1999) 590–599
2. Fox, D., Thrun, S., Burgard, W., Dellaert, F.: Particle filters for mobile robot localization. In Doucet, A., de Freitas, N., Gordon, N., eds.: *Sequential Monte Carlo Methods in Practice*, New York, Springer (2001)
3. Torma, P., Szepesvári, C.: LS-N-IPS: an improvement of particle filters by means of local search. *Proc. Non-Linear Control Systems(NOLCOS'01)* St. Petersburg, Russia (2001)
4. Liu, J.S., Chen, R.: Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association* **93** (1998) 1032–1044
5. Liu, J.S.: Metropolized independent sampling. *Statistics and Computing* **6** (1996) 113–119
6. Kong, A.: A note on importance sampling using renormalized weights. Technical report (1992)
7. Rubin, D.: Using the SIR algorithm to simulate posterior distributions. In J.M. Bernardo, M.H. DeGroot, D.L.e.A.S., ed.: *Bayesian Statistics 3*. Oxford University Press (1988) 395–402
8. Isard, M., Blake, A.: CONDENSATION – conditional density propagation for visual tracking. *International Journal Computer Vision* **29** (1998) 5–28

A Boosted Particle Filter: Multitarget Detection and Tracking

Kenji Okuma, Ali Taleghani, Nando de Freitas,
James J. Little, and David G. Lowe

University of British Columbia, Vancouver B.C V6T 1Z4, CANADA

`okumak@cs.ubc.ca`

<http://www.cs.ubc.ca/~okumak>

Abstract. The problem of tracking a varying number of non-rigid objects has two major difficulties. First, the observation models and target distributions can be highly non-linear and non-Gaussian. Second, the presence of a large, varying number of objects creates complex interactions with overlap and ambiguities. To surmount these difficulties, we introduce a vision system that is capable of learning, detecting and tracking the objects of interest. The system is demonstrated in the context of tracking hockey players using video sequences. Our approach combines the strengths of two successful algorithms: mixture particle filters and Adaboost. The mixture particle filter [17] is ideally suited to multi-target tracking as it assigns a mixture component to each player. The crucial design issues in mixture particle filters are the choice of the proposal distribution and the treatment of objects leaving and entering the scene. Here, we construct the proposal distribution using a mixture model that incorporates information from the dynamic models of each player and the detection hypotheses generated by Adaboost. The learned Adaboost proposal distribution allows us to quickly detect players entering the scene, while the filtering process enables us to keep track of the individual players. The result of interleaving Adaboost with mixture particle filters is a simple, yet powerful and fully automatic multiple object tracking system.

1 Introduction

Automated tracking of multiple objects is still an open problem in many settings, including car surveillance [10], sports [12,13] and smart rooms [6] among many others [5,7,11]. In general, the problem of tracking visual features in complex environments is fraught with uncertainty [6]. It is therefore essential to adopt principled probabilistic models with the capability of learning and detecting the objects of interest. In this work, we introduce such models to attack the problem of tracking a varying number of hockey players on a sequence of digitized video from TV.

Over the last few years, particle filters, also known as condensation or sequential Monte Carlo, have proved to be powerful tools for image tracking [3, 8,14,15]. The strength of these methods lies in their simplicity, flexibility, and systematic treatment of nonlinearity and non-Gaussianity.

Various researchers have attempted to extend particle filters to multi-target tracking. Among others, Hue *et. al* [5] developed a system for multitarget tracking by expanding the state dimension to include component information, assigned by a Gibbs sampler. They assumed a fixed number of objects. To manage a varying number of objects efficiently, it is important to have an automatic detection process. The Bayesian Multiple-Blob tracker (BraMBLe) [7] is an important step in this direction. BraMBLe has an automatic object detection system that relies on modeling a fixed background. It uses this model to identify foreground objects (targets). In this paper, we will relax this assumption of a fixed background in order to deal with realistic TV video sequences, where the background changes.

As pointed out in [17], particle filters may perform poorly when the posterior is multi-modal as the result of ambiguities or multiple targets. To circumvent this problem, Vermaak *et al* introduce a mixture particle filter (MPF), where each component (mode or, in our case, hockey player) is modelled with an individual particle filter that forms part of the mixture. The filters in the mixture interact only through the computation of the importance weights. By distributing the resampling step to individual filters, one avoids the well known problem of sample depletion, which is largely responsible for loss of track [17].

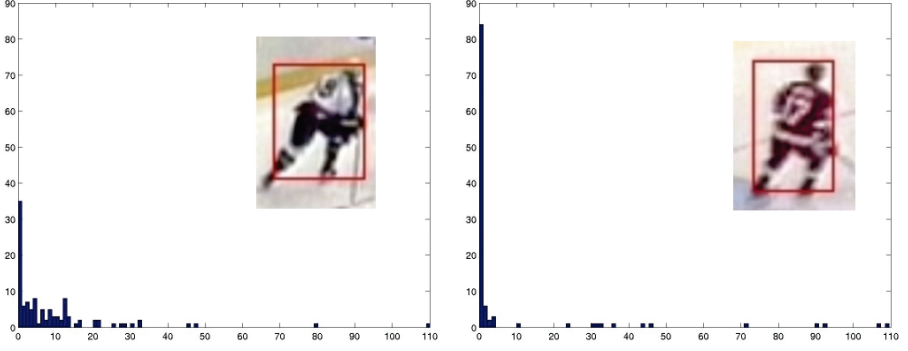
In this paper, we extend the approach of Vermaak *et al*. In particular, we use a cascaded Adaboost algorithm [18] to learn models of the hockey players. These detection models are used to guide the particle filter. The proposal distribution consists of a probabilistic mixture model that incorporates information from Adaboost and the dynamic models of the individual players. This enables us to quickly detect and track players in a dynamically changing background, despite the fact that the players enter and leave the scene frequently. We call the resulting algorithm the Boosted Particle Filter (BPF).

2 Statistical Model

In non-Gaussian state-space models, the state sequence $\{\mathbf{x}_t; t \in \mathbb{N}\}$, $\mathbf{x}_t \in \mathbb{R}^{n_{\mathbf{x}}}$, is assumed to be an unobserved (hidden) Markov process with initial distribution $p(\mathbf{x}_0)$ and transition distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, where $n_{\mathbf{x}}$ is the dimension of the state vector. In our tracking system, this transition model corresponds to a standard autoregressive dynamic model. The observations $\{\mathbf{y}_t; t \in \mathbb{N}^*\}$, $\mathbf{y}_t \in \mathbb{R}^{n_{\mathbf{y}}}$, are conditionally independent given the process $\{\mathbf{x}_t; t \in \mathbb{N}\}$ with marginal distribution $p(\mathbf{y}_t|\mathbf{x}_t)$, where $n_{\mathbf{y}}$ is the dimension of the observation vector.

2.1 Observation Model

Following [14], we adopt a multi-color observation model based on Hue-Saturation-Value (HSV) color histograms. Since HSV decouples the intensity (i.e., value) from color (i.e., hue and saturation), it is reasonably insensitive to illumination effects. An HSV histogram is composed of $N = N_h N_s + N_v$ bins and we denote $b_t(\mathbf{d}) \in \{1, \dots, N\}$ as the bin index associated with the



Color histogram of a player (LEFT: white uniform RIGHT: red uniform)

Fig. 1. Color histograms: This figure shows two color histograms of selected rectangular regions, each of which is from a different region of the image. The player on left has uniform whose color is the combination of dark blue and white and the player on right has a red uniform. One can clearly see concentrations of color bins due to limited number of colors. In (a) and (b), we set the number of bins, $N = 110$, where N_h , N_s , and N_v are set to 10

color vector $\mathbf{y}_t(\mathbf{k})$ at a pixel location \mathbf{d} at time t . Figure 1 shows two instances of the color histogram. If we define the candidate region in which we formulate the HSV histogram as $R(\mathbf{x}_t) \triangleq \mathbf{I}_t + s_t W$, then a kernel density estimate $\mathbf{K}(\mathbf{x}_t) \triangleq \{k(n; \mathbf{x}_t)\}_{n=1, \dots, N}$ of the color distribution at time t is given by [1,14]:

$$k(n; \mathbf{x}_t) = \eta \sum_{\mathbf{d} \in R(\mathbf{x}_t)} \delta[b_t(\mathbf{d}) - n] \quad (1)$$

where δ is the delta function, η is a normalizing constant which ensures k to be a probability distribution, $\sum_{n=1}^N k(n; \mathbf{x}_t) = 1$, and a location \mathbf{d} could be any pixel location within $R(\mathbf{x}_t)$. Eq. (1) defines $k(n; \mathbf{x}_t)$ as the probability of a color bin n at time t .

If we denote $\mathbf{K}^* = \{k^*(n; \mathbf{x}_0)\}_{n=1, \dots, N}$ as the reference color model and $\mathbf{K}(\mathbf{x}_t)$ as a candidate color model, then we need to measure the data likelihood (i.e., similarity) between \mathbf{K}^* and $\mathbf{K}(\mathbf{x}_t)$. As in [1,14], we apply the Bhattacharyya similarity coefficient to define a distance ξ on HSV histograms. The mathematical formulation of this measure is given by [1]:

$$\xi[\mathbf{K}^*, \mathbf{K}(\mathbf{x}_t)] = \left[1 - \sum_{n=1}^N \sqrt{k^*(n; \mathbf{x}_0) k(n; \mathbf{x}_t)} \right]^{\frac{1}{2}} \quad (2)$$

Statistical properties of near optimality and scale invariance presented in [1] ensure that the Bhattacharyya coefficient is an appropriate choice of measuring similarity of color histograms. Once we obtain a distance ξ on the HSV color histograms, we use the following likelihood distribution given by [14]:

$$p(\mathbf{y}_t | \mathbf{x}_t) \propto e^{-\lambda \xi^2[\mathbf{K}^*, \mathbf{K}(\mathbf{x}_t)]} \quad (3)$$

where $\lambda = 20$. λ is suggested in [14,17], and confirmed also on our experiments. Also, we set the size of bins N_h , N_s , and N_v as 10.

The HSV color histogram is a reliable approximation of the color density on the tracked region. However, a better approximation is obtained when we consider the spatial layout of the color distribution. If we define the tracked region as the sum of r sub-regions $R(\mathbf{x}_t) = \sum_{j=1}^r R_j(\mathbf{x}_t)$, then we apply the likelihood as the sum of the reference histograms $\{k_j^*\}_{j=1,\dots,r}$ associated with each sub-region by [14]:

$$p(\mathbf{y}_t|\mathbf{x}_t) \propto e^{\sum_{j=1}^r -\lambda \xi^2[\mathbf{K}_j^*, \mathbf{K}_j(\mathbf{x}_t)]} \quad (4)$$

Eq. (4) shows how the spatial layout of the color is incorporated into the data likelihood. In Figure 2, we divide up the tracked regions into two sub-regions in order to use spatial information of the color in the appearance of a hockey player.

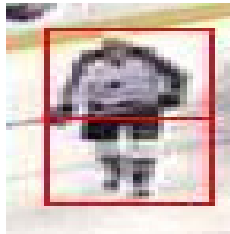


Fig. 2. Multi-part color likelihood model: This figure shows our multi-part color likelihood model. We divide our model into two sub-regions and take a color histogram from each sub-region so that we take into account the spatial layout of colors of two sub-regions

For hockey players, their uniforms usually have a different color on their jacket and their pants and the spatial relationship of different colors becomes important.

2.2 The Filtering Distribution

We denote the state vectors and observation vectors up to time t by $\mathbf{x}_{0:t} \triangleq \{\mathbf{x}_0 \dots \mathbf{x}_t\}$ and $\mathbf{y}_{0:t}$. Given the observation and transition models, the solution to the filtering problem is given by the following Bayesian recursion [3]:

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{0:t}) &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{0:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})d\mathbf{x}_{t-1}}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})d\mathbf{x}_t} \end{aligned} \quad (5)$$

To deal with multiple targets, we adopt the mixture approach of [17]. The posterior distribution, $p(\mathbf{x}_t|\mathbf{y}_{0:t})$, is modelled as an M -component non-parametric mixture model:

$$p(\mathbf{x}_t|\mathbf{y}_{0:t}) = \sum_{j=1}^M \Pi_{j,t} p_j(\mathbf{x}_t|\mathbf{y}_{0:t}) \quad (6)$$

where the mixture weights satisfy $\sum_{m=1}^M \Pi_{m,t} = 1$. Using the the filtering distribution, $p_j(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})$, computed in the previous step, the predictive distribution becomes

$$p(\mathbf{x}_t|\mathbf{y}_{0:t-1}) = \sum_{j=1}^M \Pi_{j,t-1} p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) \quad (7)$$

where $p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p_j(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1}) d\mathbf{x}_{t-1}$. Hence, the updated posterior mixture takes the form

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{0:t}) &= \frac{\sum_{j=1}^M \Pi_{j,t-1} p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{\sum_{k=1}^M \Pi_{k,t-1} \int p_k(\mathbf{y}_t|\mathbf{x}_t) p_k(\mathbf{x}_t|\mathbf{y}_{0:t-1}) d\mathbf{x}_t} \\ &= \sum_{j=1}^M \left[\frac{\Pi_{j,t-1} \int p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) d\mathbf{x}_t}{\sum_{k=1}^M \Pi_{k,t-1} \int p_k(\mathbf{y}_t|\mathbf{x}_t) p_k(\mathbf{x}_t|\mathbf{y}_{0:t-1}) d\mathbf{x}_t} \right] \\ &\quad \times \left[\frac{p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{\int p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) d\mathbf{x}_t} \right] \\ &= \sum_{j=1}^M \Pi_{j,t} p_j(\mathbf{x}_t|\mathbf{y}_{0:t}) \end{aligned} \quad (8)$$

where the new weights (independent of \mathbf{x}_t) are given by:

$$\Pi_{j,t} = \left[\frac{\Pi_{j,t-1} \int p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) d\mathbf{x}_t}{\sum_{k=1}^M \Pi_{k,t-1} \int p_k(\mathbf{y}_t|\mathbf{x}_t) p_k(\mathbf{x}_t|\mathbf{y}_{0:t-1}) d\mathbf{x}_t} \right]$$

Unlike a mixture particle filter by [17], we have M different likelihood distributions, $\{p_j(\mathbf{y}_t|\mathbf{x}_t)\}_{j=1\dots M}$. When one or more new objects appear in the scene, they are detected by Adaboost and automatically initialized with an observation model. Using a different color-based observation model allows us to track different colored objects.

3 Particle Filtering

In standard particle filtering, we approximate the posterior $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ with a Dirac measure using a finite set of N particles $\{\mathbf{x}_t^i\}_{i=1\dots N}$. To accomplish this, we sample candidate particles from an appropriate proposal distribution $\tilde{\mathbf{x}}_t^i \sim q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t})$ (In the simplest scenario, it is set as $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t}) =$

$p(\mathbf{x}_t|\mathbf{x}_{t-1})$, yielding the bootstrap filter [3]) and weight these particles according to the following importance ratio:

$$w_t^i = w_{t-1}^i \frac{p(\mathbf{y}_t|\tilde{\mathbf{x}}_t^i)p(\tilde{\mathbf{x}}_t^i|\mathbf{x}_{t-1}^i)}{q(\tilde{\mathbf{x}}_t^i|\mathbf{x}_{0:t-1}^i, \mathbf{y}_{0:t})} \quad (9)$$

We resample the particles using their importance weights to generate an un-weighted approximation of $p(\mathbf{x}_t|\mathbf{y}_{0:t})$. In the mixture approach of [17], the particles are used to obtain the following approximation of the posterior distribution:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{j=1}^M \Pi_{j,t} \sum_{i \in \mathcal{I}_j} w_t^i \delta_{\mathbf{x}_t^i}(\mathbf{x}_t)$$

where \mathcal{I}_j is the set of indices of the particles belonging to the j -th mixture component. As with many particle filters, the algorithm simply proceeds by sampling from the transition priors (note that this proposal does not use the information in the data) and updating the particles using importance weights derived from equation (8); see Section 3 of [17] for details.

In [17], the mixture representation is obtained and maintained using a simple K-means spatial reclustering algorithm. In the following section, we argue that boosting provides a more satisfactory solution to this problem.

4 Boosted Particle Filter

The boosted particle filter introduces two important extensions of the MPF. First, it uses Adaboost in the construction of the proposal distribution. This improves the robustness of the algorithm substantially. It is widely accepted that proposal distributions that incorporate the recent observations (in our case, through the Adaboost detections) outperform naive transition prior proposals considerably [15,16]. Second, Adaboost provides a mechanism for obtaining and maintaining the mixture representation. This approach is again more powerful than the naive K-means clustering scheme used for this purpose in [17]. In particular, it allows us to detect objects leaving and entering the scene efficiently.

4.1 Adaboost Detection

We adopt the cascaded Adaboost algorithm of Viola and Jones [18], originally developed for detecting faces. In our experiments, a 23 layer cascaded classifier is trained to detect hockey players. In order to train the detector, a total of 6000 figures of hockey players are used. These figures are scaled to have a resolution of 10×24 pixels. In order to generate such data in a limited amount of time, we speed the selection process by using a program to extract small regions of the image that most likely contain hockey players. Simply, the program finds regions that are centered on low intensities (i.e., hockey players) and surrounded by high intensities (i.e., rink surface). However, it is important to note that the



Fig. 3. Training set of images for hockey players: This figure shows a part of the training data. A total of 6000 different figures of hockey players are used for the training

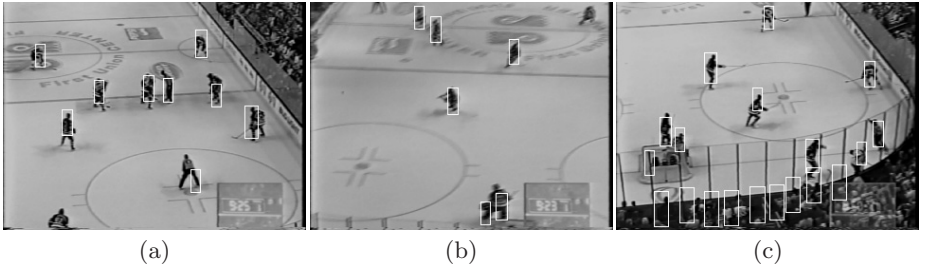


Fig. 4. Hockey player detection result: This figure shows results of the Adaboost hockey detector. (a) and (b) shows mostly accurate detections. In (c), there are a set of false positives detected on audience by the rink

data generated by such a simple script is not ideal for training Adaboost, as shown in Figure 3. As a result, our trained Adaboost produces false positives alongside the edge of the rink shown in (c) of Figure 4. More human intervention with a larger training set would lead to better Adaboost results, although failures would still be expected in regions of clutter and overlap. The non-hockey-player subwindows used to train the detector are generated from 100 images manually chosen to contain nothing but the hockey rink and audience. Since our tracker is implemented for tracking hockey scenes, there is no need to include training images from outside the hockey domain.

The results of using Adaboost in our dataset are shown in Figure 4. Adaboost performs well at detecting the players but often gets confused and leads to many false positives.

4.2 Incorporating Adaboost in the Proposal Distribution

It is clear from the Adaboost results that they could be improved if we considered the motion models of the players. In particular, by considering plausible motions, the number of false positives could be reduced. For this reason, we in-

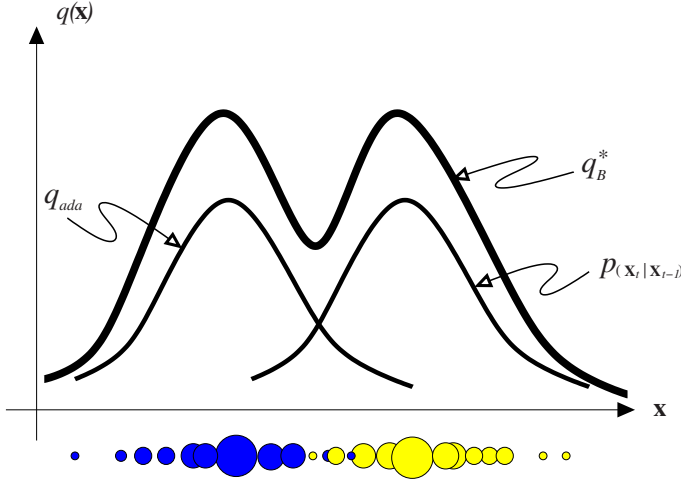


Fig. 5. Mixture of Gaussians for the proposal distribution

corporate Adaboost in the proposal mechanism of our MPF. The expression for the proposal distribution is given by the following mixture.

$$q_B^*(\mathbf{x}_t | \mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) = \alpha q_{ada}(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_t) + (1 - \alpha) p(\mathbf{x}_t | \mathbf{x}_{t-1}) \quad (10)$$

where q_{ada} is a Gaussian distribution that we discuss in the subsequent paragraph (See Figure 5). The parameter α can be set dynamically without affecting the convergence of the particle filter (it is only a parameter of the proposal distribution and therefore its influence is corrected in the calculation of the importance weights). When $\alpha = 0$, our algorithm reduces to the MPF of [17]. By increasing α we place more importance on the Adaboost detections. We can adapt the value of α depending on tracking situations, including cross overs, collisions and occlusions.

Note that the Adaboost proposal mechanism depends on the current observation \mathbf{y}_t . It is, therefore, robust to peaked likelihoods. Still, there is another critical issue to be discussed: determining how close two different proposal distributions need to be for creating their mixture proposal. We can always apply the mixture proposal when q_{ada} is overlaid on a transition distribution modeled by autoregressive state dynamics. However, if these two different distributions are not overlapped, there is a distance between the mean of these distributions.

If a Monte Carlo estimation of a mixture component by a mixture particle filter overlaps with the nearest cluster given by the Adaboost detection algorithm, we sample from the mixture proposal distribution. If there is no overlap between the Monte Carlo estimation of a mixture particle filter for each mixture component and clusters given by the Adaboost detection, then we set $\alpha = 0$ so that our proposal distribution takes only a transition distribution of a mixture particle filter.

5 Experiments

This section shows BPF tracking results on hockey players in a digitized video sequence from broadcast television. The experiments are performed using our non-optimized implementation in C on a 2.8 GHz Pentium IV.

Figure 6 shows scenes where objects are coming in and out. In this figure, it is important to note that no matter how many objects are in the scene, the mixture representation of BPF is not affected and successfully adapts to the change. For objects coming into the scene, in (a) of the figure, Adaboost quickly detects a new object in the scene within a short time sequence of only two frames. Then BPF immediately assigns particles to an object and starts tracking it.

Figure 7 shows the Adaboost detection result in the left column, a frame number in the middle, and BPF tracking results on the right. In Frame 32 and 33, a small crowd of three players in the middle of the center circle on the rink are not well detected by Adaboost. This is an example of the case

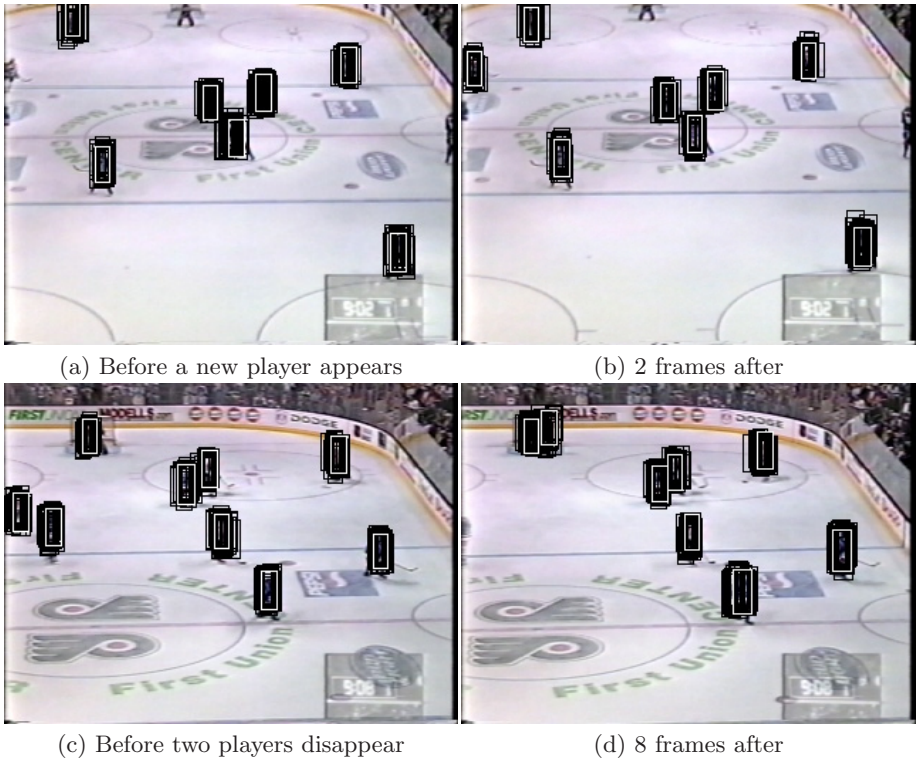


Fig. 6. Objects appearing and disappearing from the scene: This is a demonstration of how well BPF handles objects randomly coming in and out of the scene. (a) and (b) show that a new player that appears in the top left corner of the image is successfully detected and starts to get tracked. (c) and (d) show that two players are disappearing from the scene to left

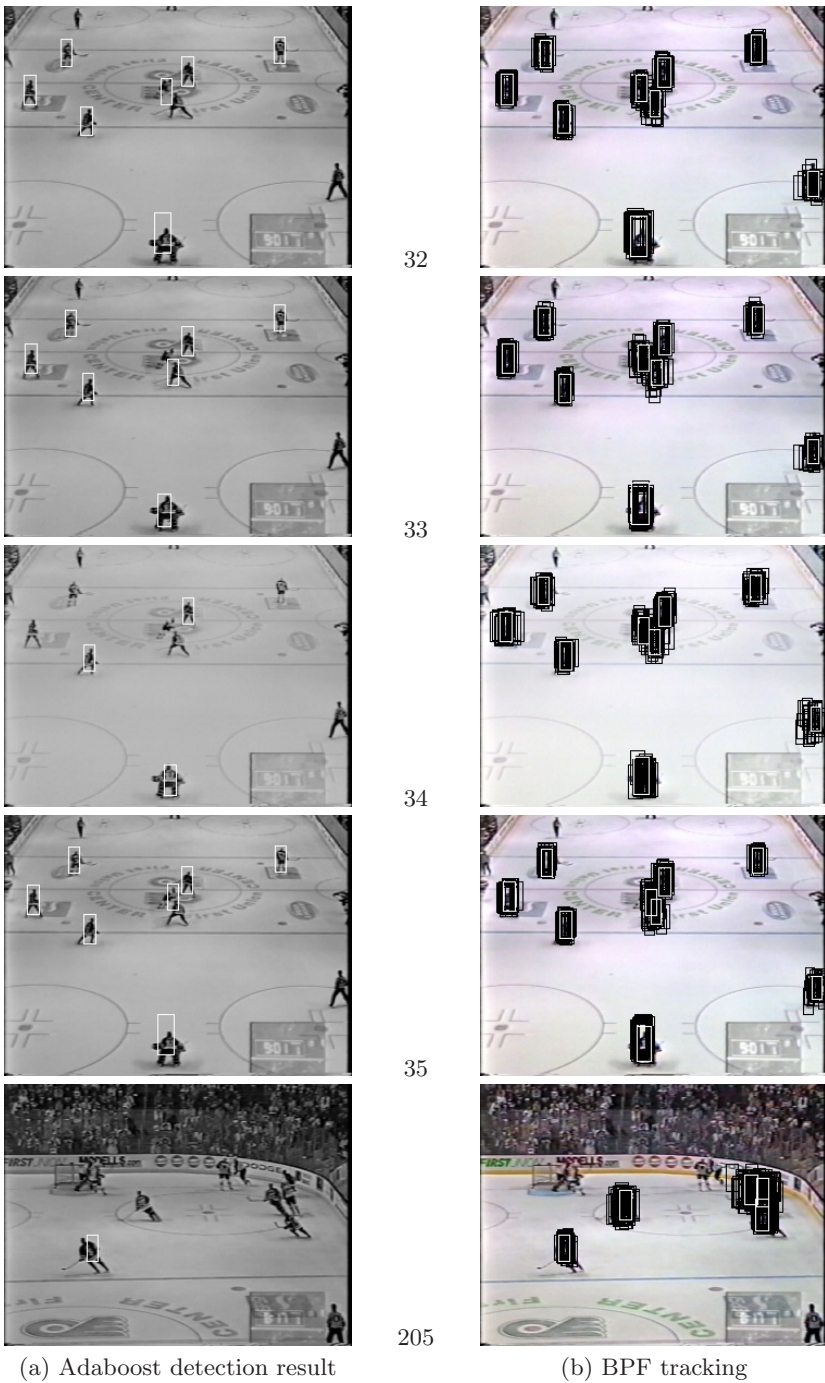


Fig. 7. BPF tracking result: The results of Adaboost detection are shown on the left, with the corresponding boosted particle filter results on the right

in which Adaboost does not work well in clutter. There are only two windows detected by Adaboost. However, BPF successfully tracks three modes in the cluttered environment. Frames 34 and 35 show another case when Adaboost fails. Since Adaboost features are based on different configurations of intensity regions, it is highly sensitive to a drastic increase/decrease of the intensity. The average intensity of the image is clearly changed by a camera flash in Frame 34. The number of Adaboost detections is much smaller in a comparison with the other two consecutive frames. However, even in the case of an Adaboost failure, mixture components are well maintained by BPF, which is also clearly shown in Frame 205 in the figure.

6 Conclusions

We have described an approach to combining the strengths of Adaboost for object detection with those of mixture particle filters for multiple-object tracking. The combination is achieved by forming the proposal distribution for the particle filter from a mixture of the Adaboost detections in the current frame and the dynamic model predicted from the previous time step. The combination of the two approaches leads to fewer failures than either one on its own, as well as addressing both detection and consistent track formation in the same framework.

We have experimented with this boosted particle filter in the context of tracking hockey players in video from broadcast television. The results show that most players are successfully detected and tracked, even as players move in and out of the scene. We believe results can be further improved by including more examples in our Adaboost training data for players that are seen against non-uniform backgrounds. Further improvements could be obtained by dynamic adjustment of the weighting parameter selecting between the Adaboost and dynamic model components of the proposal distribution. Adopting a probabilistic model for target exclusion [11] may also improve our BPF tracking.

Acknowledgment. This research is funded by the Institute for Robotics and Intelligent Systems (IRIS) and their support is gratefully acknowledged. The authors would like to specially thank Matthew Brown, Jesse Hoey, and Don Murray from the University of British Columbia for fruitful discussions and helpful comments about the formulation of BPF.

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Real-Time Tracking of Non-Rigid Objects using Mean Shift. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 142-151 (2000)
2. Deutscher, J., Blake, A., Ried, I.: Articulated body motion capture by annealed particle filtering. *IEEE Conference on Computer Vision and Pattern Recognition*, (2000)
3. Doucet, A., de Freitas, J. F. G., N. J. Gordon, editors: *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York (2001)
4. Freund, Y., Schapire, R. E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Computational Learning Theory*, pp. 23-37, Springer-Verlag, (1995)
5. Hue, C., Le Cadre, J.-P., Pérez, P.: Tracking Multiple Objects with Particle Filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):791-812 (2002)
6. Intille, S. S., Davis, J. W., Bobick, A.F.: Real-Time Closed-World Tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 697-703 (1997)
7. Isard, M., MacCormick, J.: BraMBLe: A Bayesian multiple-blob tracker. *International Conference on Computer Vision*, pp. 34-41(2001)
8. Isard, M., Blake, A.: Condensation – conditional density propagation for visual tracking. *International Journal on Computer Vision*, 28(1):5-28 (1998)
9. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems *Transactions of the ASME–Journal of Basic Engineering*, vol.82 Series D pp.35-45 (1960)
10. Koller, D., Weber, J., Malik, J.: Robust Multiple Car Tracking with Occlusion Reasoning. *European Conference on Computer Vision*, pp. 186-196, LNCS 800, Springer-Verlag (1994)
11. MacCormick, J., Blake, A.: A probabilistic exclusion principle for tracking multiple objects. *International Conference on Computer Vision*, pp. 572-578 (1999)
12. Misu, T., Naemura, M., Wentao Zheng, Izumi, Y., Fukui, K.: Robust Tracking of Soccer Players Based on Data Fusion *IEEE 16th International Conference on Pattern Recognition*, pp. 556-561 vol.1 (2002)
13. Needham, C. J., Boyle, R. D.: Tracking multiple sports players through occlusion, congestion and scale. *British Machine Vision Conference*, vol. 1, pp. 93-102 *BMVA* (2001)
14. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. *European Conference on Computer Vision*, (2002)
15. Rui, Y., Chen, Y.: Better Proposal Distributions: Object Tracking Using Unscented Particle Filter. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 786-793 (2001)
16. van der Merwe, R., Doucet, A., de Freitas, J. F. G., Wan, E: The Unscented Particle Filter. *Advances in Neural Information Processing Systems*, vol. 8 pp 351-357 (2000)
17. Vermaak, J., Doucet, A., Pérez, P.: Maintaining Multi-Modality through Mixture Tracking. *International Conference on Computer Vision* (2003)
18. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. *IEEE Conference on Computer Vision and Pattern Recognition* (2001)

Simultaneous Object Recognition and Segmentation by Image Exploration^{*}

Vittorio Ferrari¹, Tinne Tuytelaars², and Luc Van Gool^{1,2}

¹ Computer Vision Group (BIWI), ETH Zuerich, Switzerland
{ferrari,vangool}@vision.ee.ethz.ch

² ESAT-PSI, University of Leuven, Belgium
Tinne.Tuytelaars@esat.kuleuven.ac.be

Abstract. Methods based on local, viewpoint invariant features have proven capable of recognizing objects in spite of viewpoint changes, occlusion and clutter. However, these approaches fail when these factors are too strong, due to the limited repeatability and discriminative power of the features. As additional shortcomings, the objects need to be rigid and only their approximate location is found. We present a novel Object Recognition approach which overcomes these limitations. An initial set of feature correspondences is first generated. The method anchors on it and then gradually explores the surrounding area, trying to construct more and more matching features, increasingly farther from the initial ones. The resulting process covers the object with matches, and simultaneously separates the correct matches from the wrong ones. Hence, recognition and segmentation are achieved at the same time. Only very few correct initial matches suffice for reliable recognition. The experimental results demonstrate the stronger power of the presented method in dealing with extensive clutter, dominant occlusion, large scale and viewpoint changes. Moreover non-rigid deformations are explicitly taken into account, and the approximative contours of the object are produced. The approach can extend any viewpoint invariant feature extractor.

1 Introduction

Recently, object recognition (OR) approaches based on local invariant features have become increasingly popular [8,5,2,4,7]. Typically, local features are extracted independently from both a model and a test image, then characterized by invariant descriptors and finally matched. The success of these approaches is twofold. First, the feature extraction process and description are viewpoint invariant. Secondly, local features bring tolerance to clutter and occlusion, de facto removing the need for prior segmentation. In this respect, global methods, both contour-based [9] and appearance-based [10], are a step behind.

In spite of their success, the robustness and generality of these approaches are limited by the repeatability of the feature extraction, and the difficulty of

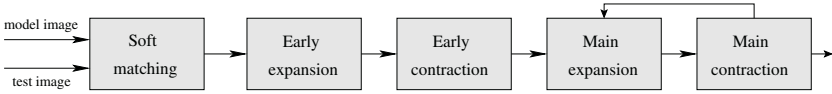
^{*} This research was supported by EC project VIBES and the Fund for Scientific Research Flanders.

matching correctly, in the presence of large amounts of clutter and challenging viewing conditions. Large scale or viewpoint changes considerably lower the probability that any given model feature is re-extracted in the test image (e.g.: figure 2, left). Simultaneously, occlusion reduces the number of visible model features. The combined effect is that only a small fraction of model features has a correspondence in the test image. This fraction represents the maximal number of features that can be correctly matched. Unfortunately, at the same time extensive clutter gives rise to a large number of non-object features, which disturb the matching process. As a final outcome of these combined difficulties, only a few, if any, correct matches are produced. Because these often come together with many mismatches, recognition tends to fail.

Even in easier cases, to suit the needs for repeatability in spite of viewpoint changes, only a sparse set of *distinguished* features [7] are extracted. As a result, only a small portion of the object is typically covered with matches. Densely covering the visible part of the object is desirable, as it increases the *evidence* for its presence, which results in higher discriminative power.

In this paper, we face these problems by no longer relying solely on matching viewpoint invariant features. Instead, we propose to anchor on an initial set thereof, and then *look around* them trying to construct more matching features. As new matches arise, they are exploited to construct even more, in a process which gradually *explores* the test image, recursively constructing more and more matches, increasingly farther from the initial ones. As the number and extent of matched features increases, so does the information available to judge their individual correctness. Gradually the system’s confidence in the presence of the object grows.

We build upon a multi-scale extension of the affine invariant region extractor of [2]. An initial large set of unreliable region correspondences is generated through a process tuned to maximize the amount of correct matches, at the cost of producing many mismatches (section 2). Additionally, we generate a grid of circular regions homogeneously covering the model image. The core of the method iteratively alternates between *expansion* phases, where correspondences for these coverage regions are constructed, and *contraction* phases, which attempt to remove mismatches. In the first expansion phase (section 3), we try to propagate the coverage regions based on the geometric transformation of nearby initial matches. By *propagating* a region, we mean constructing the corresponding one in the test image. The propagated matches and the initial ones are then passed through a novel local filter, during the first contraction phase (section 4). The processing continues by alternating faster expansion phases (section 5), where coverage regions are propagated over a larger area, with contraction phases based on a global filter (section 6). The filter exploits both topological arrangements and appearance information, and tolerates *non-rigid deformations*. During the expansion phases, the shape of each new region is adapted to the local surface orientation, thus allowing the exploration process to follow curved surfaces and deformations (e.g. a folded magazine). At each iteration, the presence of the newly propagated matches helps the filter to take better removal decisions. In turn, the cleaner set of supports makes the next expansion more effective. As a

**Fig. 1.** Scheme of the system

result, the amount, and the percentage, of correct matches grows every iteration. The algorithm is getting a clearer idea about the object's presence and location. The two closely cooperating processes of expansion and contraction gather more evidence about the presence of the object and separate correct matches from wrong ones *at the same time*. This results in the simultaneous recognition and segmentation of the object. By constructing matches for the coverage regions, the system succeeds in covering also image areas which are not interesting for the feature extractor or not discriminative enough to be correctly matched by traditional techniques.

The basic advantage of the approach is that each single correct initial match can expand to cover a contiguous surface with *many* correct matches, even when starting from a large amount of mismatches. This leads to filling the visible portion of the object with matches. Some interesting *direct* advantages derive from it. First, robustness to scale, viewpoint, occlusion and clutter are greatly enhanced, because most cases where the traditional approach generated only a few correct matches are now solvable. Second, discriminative power is increased, because decisions about the object's identity are based on information densely distributed over the entire portion of the object visible in the test image. Third, the approximate boundary of the object in the test image is directly suggested by the final set of matched regions (section 8). Fourth, non-rigid deformations are explicitly taken into account.

2 Soft Matches

The feature extraction algorithm [2] is applied to both a *model image* I_m and a *test image* I_t independently, producing two sets of regions Φ_m, Φ_t .

Tentative Matches

For each test region $T \in \Phi_t$ we compute the Mahalanobis distance of the invariant descriptors [2] to all model regions $M \in \Phi_m$. An appearance similarity measure $\text{sim}(T, M)$ is computed between T and each of the 10 closest regions. The measure is a linear combination of grey-level normalized cross-correlation (NCC) and the average Euclidean distance in *RGB* space, after geometric and photometric normalization. This mixture is more discriminant than NCC alone, while keeping invariance to brightness changes. We consider each of the 3 most similar regions above a low threshold t_1 . Repeating this operation for all regions $T \in \Phi_t$, yields a first set of *tentative matches*. At this point, every test region could be matched to either none, 1, 2 or 3 model regions.

Refinement and Re-thresholding

Since all regions are independently extracted from the two images, the geometric registration of a correct match might not be optimal, which lowers its similarity. The registration of the tentative matches is *refined* using our recently proposed algorithm [1], that efficiently looks for the affine transformation that maximizes the similarity. After refinement, the similarity is re-evaluated and only matches scoring above a second, higher threshold t_2 are kept. Refinement tends to raise the similarity of correct matches much more than that of mismatches. The increased *separation* between the similarity distributions makes the second thresholding more effective.

The obtained set of matches usually still contains *soft-matches*, i.e. more than one region in Φ_m corresponding to the same region in Φ_t , or vice-versa. This contrasts with classic matching methods [7,2,5,11,8], but there are two good reasons for it. First, the scene might contain repeated, or visually similar elements. Secondly, large viewpoint and scale changes cause loss of resolution which results in a less accurate correspondence and a lower similarity. When there is also extensive clutter, it might be impossible, based *purely* on local appearance [14], to decide which of the top-3-matches is correct, as several competing regions might appear very similar, and score higher than the correct match.

The proposed process outputs a large set of plausible matches, all with a reasonably high similarity. The goal is to maximize the amount of correct matches, even at the cost of accepting a substantial fraction of mismatches. In difficult



Fig. 2. Left: case-study (top: model image, bottom: test image). Middle: a closer view with 3 initial matches. The two model regions on the left are both matched to the same region in the test image. Note the small occluding rubber on the spoon. Right-top: the homogeneous coverage Ω . Right-bottom: a support region (dark), associated sectors (lines) and candidates (bright)

cases this is important, as each correct match can start an expansion which will cover significant parts of the object.

Figure 2 shows a case-study, for which 3 correct matches out of 217 are found (a *correct-ratio* of 3/217). The large scale change (factor 3.3), combined with the modest resolution (720x576), causes heavy image degradation which corrupts edges and texture. In such conditions only a few model regions are re-extracted and many mismatches are inevitable. In the remainder of the paper, we refer to the current set of matches as the *configuration* Γ .

How to proceed ? Global, robust geometry filtering methods, like detecting outliers to the epipolar geometry through RANSAC [3] fail, as they need a minimal amount of inliers of about 30% [8]. Initially, this may very well not be the case. Even if we could separate out the few correct matches, they would not be sufficient to draw reliable conclusions about the presence of the object. In the following we explain how to gradually increment the number of correct matches and simultaneously decrease the number of mismatches.

3 Early Expansion

Coverage of the Model Image

We generate a grid Ω of overlapping circular regions densely covering the model image I_m (figure 2, top-right). The expansion phases will try to construct in I_t as many regions corresponding to them as possible.

Propagation Attempt

We now define the concept of *propagation attempt* which is the basic building-block of the expansion phases and will be used later. Consider a region C_m in model image I_m without match in the test image I_t and a nearby region S_m , matched to S_t . If C_m and S_m lie on the same physical facet of the object, they will be mapped to I_t by similar affine transformations. The *support* match (S_m, S_t) *attempts to propagate* the *candidate* region C_m to I_t as follows:

1. Compute the affine transformation A mapping S_m to S_t .
2. Project C_m to I_t via $A : C_t = AC_m$.

The benefits of exploiting previously established geometric transformations was also noted by [13].

Early Expansion

Propagation attempts are used as follows. Consider as supports $\{S^i = (S_m^i, S_t^i)\}$ the soft-matches configuration Γ , and as candidates Λ the coverage regions Ω . For each support region S_m^i we partition I_m into 6 circular sectors centered on the center of S_m^i (figure 2, bottom-right). Each S_m^i attempts to propagate the closest candidate region in each sector. As a consequence, each candidate C_m has an associated subset $\Gamma_{C_m} \subset \Gamma$ of supports that will *compete* to propagate it. For a candidate C_m and each support S^i in Γ_{C_m} do:

1. Generate C_t^i by attempting to propagate C_m via S^i .
2. Refine C_t^i . If C_t^i correctly matches C_m , this adapts it to the local surface orientation (handles curved and deformable objects) and perspective effects (the affine approximation is only valid on a local scale).
3. Evaluate the quality of the refined propagation attempt: $sim_i = \text{sim}(C_m, C_t^i)$

We retain C_t^{best} , with $best = \arg \max_i sim_i$, the best refined propagation attempt. C_m is considered successfully propagated to C_t^{best} if $sim_{best} > t_2$ (the matching threshold). This procedure is applied for all candidates $C_m \in \Lambda$.

Most support matches may actually be mismatches, and many of them typically lie around each of the few correct ones (e.g.: several matches in a single soft-match, figure 2, middle). In order to cope with this situation, each support concentrates its efforts on the nearest candidate in each direction, as it has the highest chance to undergo a similar geometric transformation. Additionally, every propagation attempt is refined before evaluation. Refinement raises the similarity of correctly propagated matches much more than the similarity of mispropagated ones, thereby helping correct supports to win. This results in a limited, but controlled growth, maximizing the chance that each correct match propagates, and limiting the proliferation of mispropagations. The process also restricts the number of refinements to at most 6 per support (contains computational cost).

For the case-study, 113 new matches are generated and added to the configuration Γ . 17 of them are correct and located around the initial 3. The correct-ratio of Γ improves to 20/330 (figure 4, left), but it is still very low.

4 Early Contraction

The early expansion guarantees high chances that each initial correct match propagates. As initial filter, we discard all matches that did not succeed in propagating any region. The correct-ratio improves to 20/175 (no correct match is lost), but it is still too low for applying a global filter. Hence, we have developed the following local filter.

A local group of regions in the model image have uniform shape, are arranged on a grid and intersect each other with a specific pattern. If all these regions are correctly matched, the same regularities also appear in the test image, because the surface is contiguous and smooth (regions at depth discontinuities can't be matched correctly anyway). This holds for curved or deformed objects as well, because the affine transformation varies slowly and smoothly across neighboring regions (figure 3, left). On the other hand, mismatches tend to be located elsewhere in the image and to have different shapes. We propose a novel, local filter based on this observation. Let $\{N_m^i\}$ be the neighbors of a region R_m in the model image. Two regions A, B are considered neighbors if they intersect, i.e.: if $\text{Area}(A \cap B) > 0$. Only neighbors which are actually matched to the test image are considered. Any match (R_m, R_t) is removed from Γ if

$$\sum_{\{N_m^i\}} \left| \frac{\text{Area}(R_m \cap N_m^i)}{\text{Area}(R_m)} - \frac{\text{Area}(R_t \cap N_t^i)}{\text{Area}(R_t)} \right| > t_s$$



Fig. 3. Left: the regular arrangement of the regions is preserved. Middle: top: a candidate (thin) and 2 of 20 supports (thick) within the large circular area. bottom: the candidate is propagated to the test image using the affine transformation of the support on the right. Refinement adapts the shape to the perspective (brighter). Right: sidedness constraint. R^1 is on the same side of the line in both images

with t_s some threshold. The filter tests the preservation of the pattern of intersections between R and its neighbors (the ratio of areas is affine invariant). Hence, a removal decision is based solely on *local* information. As a consequence, this filter is unaffected by the current, low overall ratio of correct matches. Shape information is integrated in the filter, making it capable of spotting insidious mismatches which are roughly correctly located, yet have a wrong shape. This is an advantage over the (semi-)local filter proposed by [6], and later also used by others [14], which verifies if a minimal amount of regions in an area around R_m in the model image also match near R_t in the test image.

The input regions need not be arranged in a regular grid, the filter applies to a general set of (intersecting) regions. Note that incorrectly matched regions with no neighbors will not be detected. The algorithm can be implemented to run in $O(|\Gamma| + x)$, with $x \ll |\Gamma|^2$ the number of region intersections.

Applying this filter to the case-study brings the correct-ratio of Γ to 13/58, thereby greatly reducing the number of mismatches.

5 Main Expansion

The first ‘early’ expansion and contraction phases brought several additional correct matches and removed many mismatches, especially those that concentrated around the correct ones. Since Γ is cleaner, we can now try a faster expansion.

All matches in the current configuration Γ are removed from the candidate set $\Lambda \leftarrow \Lambda \setminus \Gamma$, and are used as supports. All support regions S_m^i in a circular area ¹ around a candidate C_m compete to propagate it:

1. Generate C_t^i by attempting to propagate C_m via S^i
2. Evaluate $\text{sim}_i = \text{sim}(C_m, C_t^i)$

We retain C_t^{best} , with $\text{best} = \arg \max_i \text{sim}_i$ and refine it, yielding C_t^{ref} . C_m is considered successfully propagated to C_t^{ref} if $\text{sim}(C_m, C_t^{\text{ref}}) > t_2$ (figure 3, middle). This scheme is applied for each candidate.

In contrast to the early expansion, many more supports compete for the same candidate, and no refinement is applied *before* choosing the winner. However, the presence of more correct supports, now tending to be grouped, and fewer mismatches, typically spread out, provides good chances that a correct support will win a competition. In this process each support has the chance to propagate many more candidates, spread over a larger area, because it offers help to all candidates within a wide circular radius. This allows the system to grow a *mass* of correct matches. Moreover, the process can jump over small occlusions or degraded areas, and costs only one refinement per candidate. 185 new matches, 61 correct, are produced for the case-study, thus lifting the correct-ratio of Γ to 74/243 (30.5%, figure 4, middle).

6 Main Contraction

At this point the chances of having a sufficient number of correct matches to try a global filter are much better. In contrast to the local filter of section 4, the following global filter is capable of finding also isolated mismatches. The algorithm extends our topological filter in [1] to include also appearance similarity.

Figure 3 (right) illustrates the property on which the filter is based. The center of a region R^1 should be on the same side of the directed line going from the center of a second region R^2 to the center of a third region R^3 in both the model and test images (noted $\text{side}(R^1, R^2, R^3)$). This *sidedness constraint* holds for all correctly matched triples of coplanar regions and also for most non-coplanar ones [1]. It does not hold for non-coplanar triples in presence of strong parallax in a few cases, coined *parallax-violations* [1].

A triple including any mismatched region has higher chances to violate the constraint. When this happens, we can only conclude that probably at least one of the matches is incorrect, but we do not yet know which. However, by integrating the weak information each triple provides, it is possible to robustly discover mismatches. Hence, we check the constraint for all unordered triples and we expect wrong matches to be involved in a higher share of violations:

$$\text{err}_{\text{topo}}(R^i) = \frac{1}{v} \sum_{R^j, R^k \in \Gamma \setminus R^i, j > k} |\text{side}(R_m^i, R_m^j, R_m^k) - \text{side}(R_t^i, R_t^j, R_t^k)| \quad (1)$$

¹ In all experiments the radius is set to 1/6 of the image size.

with $v = (n - 1)(n - 2)/2$, $n = |I|$. $\text{err}_{\text{topo}}(R^i) \in [0, 1]$ because it is normalized w.r.t. the maximum number of violations v any region can be involved in. As a novel extension to [1], the topological error share (1) is combined with an appearance term, giving the total error

$$\text{err}_{\text{tot}}(R^i) = \text{err}_{\text{topo}}(R^i) + (t_2 - \text{sim}(R_m^i, R_t^i))$$

The filtering algorithm goes as follows:

1. (Re-)compute $\text{err}_{\text{tot}}(R^i)$ for all $R^i \in I$.
2. Find the worst match R^w , with $w = \arg \max_i \text{err}_{\text{tot}}(R^i)$
3. If $\text{err}_{\text{tot}}(R^w) > 0$, remove R^w : $I \leftarrow (I \setminus R^w)$, and iterate to 1, else stop.

The idea of the algorithm is that at each iteration the most probable mismatch R^w is removed and the error of correct matches decreases, because they are involved in less triples containing any mismatch. After several iterations, ideally only correct matches are left and the algorithm stops. The second term of err_{tot} decreases with increasing appearance similarity, and it vanishes when $\text{sim}(R_m^i, R_t^i) = t_2$, the matches acceptance threshold. The removal criteria $\text{err}_{\text{tot}} > 0$ expresses the idea that topological violations are accepted up to the degree to which they are compensated by high similarity. This helps finding mismatches which can hardly be judged by only one cue. A typical mismatch with similarity just above t_2 , will be removed unless it is perfectly topologically located. Conversely, correct matches with $\text{err}_{\text{topo}} > 0$ due to parallax-violations are in little danger, because they typically have good similarity. Including appearance makes the filter more robust to low correct-ratios, and remedies the drawback (parallax-violations) of the purely topological filter [1].

The proposed method offers two main advantages over rigid-motion filters, traditionally used in the matching literature [2,5,4,13,7,14], e.g.: detecting outliers to the epipolar geometry through RANSAC [3]. First, it allows for non-rigid deformations, like the bending of paper or cloth, because the structure of the spatial arrangements, captured by the sidedness constraints, is stable under these transformations. Second, it is much less sensitive to inaccurate localizations, because err_{topo} varies slowly and smoothly for a region departing from its ideal location.

Topological configurations of points and lines are also used in [15], which enforces the cyclic ordering of line segments connecting corners as a mean for steering the matching process.

In the case-study, the filter starts from 74/243 and returns 54/74, which is a major improvement. 20 correct matches are lost, but many more mismatches (149) are removed. The further processing will recover the lost correct matches and generate even more.

7 Exploring the Test Image

The processing continues by iteratively alternating main expansion and main contraction phases:

1. Do a main expansion phase. This produces a set of propagated region matches \mathcal{R} , which are added to the configuration: $\Gamma \leftarrow (\Gamma \cup \mathcal{R})$.
2. Do a main contraction phase on Γ .
3. If at least one newly propagated region survives the contraction, i.e. $|\mathcal{R} \cap \Gamma| > 0$, then iterate to 1, after updating the candidate set to contain $\Lambda \leftarrow (\Omega \setminus \Gamma)$, all original candidate regions Ω which are not yet in the configuration.

In the first iteration, the expansion phase generates some correct matches, along with some mismatches, thereby increasing the correct-ratio. The first main contraction phase removes mostly mismatches, but might also lose several correct matches: the amount of noise could still be high and limit the filter's performance. In the next iteration, this cleaner configuration is fed into the expansion phase again which, less distracted, generates more correct matches and fewer mismatches. The new correct matches in turn help the next contraction stage in taking better removal decisions, and so on. As a result, the amount, percentage and spatial extent of correct matches increase at every iteration, reinforcing the confidence about the object's presence and location. The two goals of separating correct matches and gathering more information about the object are achieved *at the same time*.

Correct matches erroneously killed by the contraction step in an iteration get another chance during the next expansion phase. With even fewer mismatches present, they are probably regenerated, and this time have higher chances to survive the contraction (higher correct-ratio, more positive evidence present).

Thanks to the refinement, each expansion phase adapts the shape of the newly created regions to the local surface orientation. Thus the whole exploration process follows curved surfaces and deformations.

The exploration procedure tends to 'implode' when the object is not in the test image, typically returning 0, or at most a few matches. Conversely, when the object is present, the approach fills the visible portion with many high confidence matches. This yields high discriminative power and the qualitative shift from only *detecting* the object to knowing its extent in the image and which parts are occluded. Recognition and segmentation are intensely intertwined.



Fig. 4. Case-study. Left: 20 correct matches (dark) out of 330 after early expansion. Middle: 74/243 after the first main expansion. Right: contour of the final set of matches. Note the segmentation quality, in particular the detection of the occluding rubber

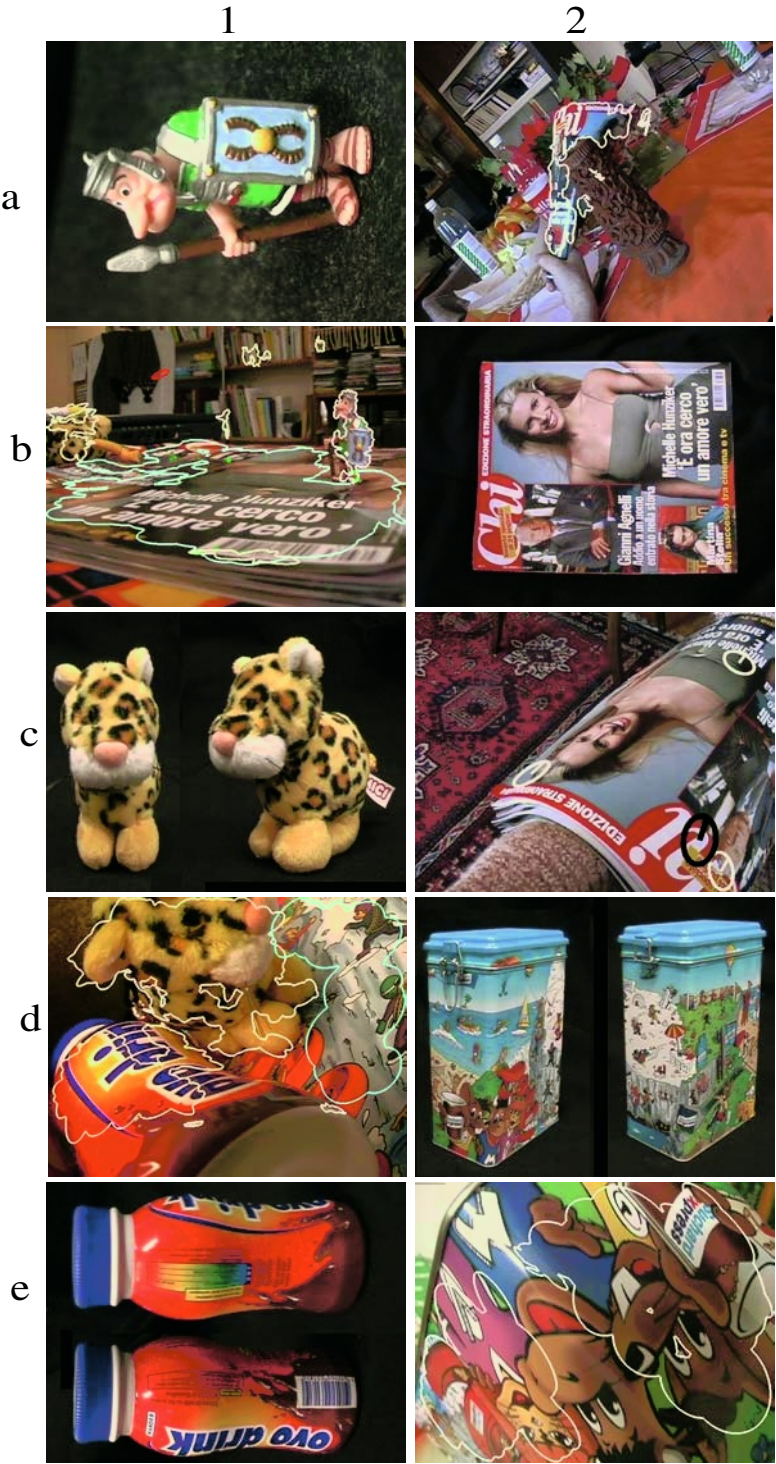
In the case-study, the second main expansion propagates 141 matches, 117 correct, which is better than the previous 61/185. The second main contraction starts from 171/215 and returns 150/174, killing a lower percentage of correct matches than the first main contraction. After the 11th iteration 220 matches cover the whole visible part of the object (figure 4, right).

8 Results and Conclusion

We report results for a set of 9 model objects and 23 test images. In total, the objects appear 43 times, as some test images contain several objects. There are 3 planar objects, each modeled by a single view, including a *Kellogs* box (figure 2), and two magazines *Michelle* (figure b2) and *Blonde* (analog model view). Two objects with curved shapes, *Xmas* (g2) and *Ovo* (e1), have 6 model views. *Leo* (c1), *Car* (f1), *Suchard* (d2) feature more complex 3D shape and have 8 model views. Finally, one frontal view models the last 3D object, *Guard* (a1). Multiple model views are taken equally spaced around the object. The contributions from all model views are integrated by superimposing the area covered by the final set of matched regions (to find the contour), and by summing their number (recognition criteria). All images are shot at a modest resolution (720x576) and all experiments are conducted with the same set of parameters. In general, in the test cases there is considerable clutter and the objects appear smaller than in the models (all models are shown at the same scale as the test images).

Tolerance to deformations is shown in a2, where *Michelle* is simultaneously strongly folded and occluded. The contours are found with a good accuracy, extending to the left until the edge of the object. Note the extensive clutter. High robustness to viewpoint changes is demonstrated in b1, where *Leo* is only half visible and captured in a considerably different pose than any of the model views, while *Michelle* undergoes a very large out-of-plane rotation of about 80 degrees. *Guard*, occluding *Michelle*, is also detected in the image, despite a scale change of factor 3. In d1, *Leo* and *Ovo* exhibit significant viewpoint change, while *Suchard* is simultaneously scaled factor 2.2 and 89% occluded. This very high occlusion level makes this case challenging even for a human observer. A scale change of factor 4 affecting *Suchard* is illustrated in e2. In figure f2, *Xmas* is divided in two by a large occludor. Both visible parts are correctly detected by the presented method. On the right size of the image, *Car* is found even if half occluded and very small. *Car* is also detected in spite of considerable viewpoint change in g1. The combined effects of strong occlusion, scale change and clutter make h2 an interesting case. Note how the boundaries of *Xmas* are accurately found, and in particular the detection of the part behind the glass. As a final example, 8 objects are detected at the same time in i2 (for clarity, only 3 contours are shown). Note the correct segmentation of the two deformed magazines and the simultaneous presence of all the aforementioned difficulty factors.

Figure h1 presents a close-up on one of 93 matches produced between a model view of *Xmas* (left) and test case h2 (right). This exemplifies the great appearance variation resulting from combined viewpoint, scale and illumination





changes, and other sources of image degradation (here a glass). In these cases, it is very unlikely for the region to be detected by the initial region extractor, and hence traditional methods fail. This figure also illustrates the accuracy of the correspondences generated by the expansion phases.

As a proof of the method's capability of following deformations, we tried to process the case in c2 starting with only one match (dark). 356 regions, covering the whole object, were produced. Each region's shape fits the local surface orientation (for clarity, only 3 regions are shown).

The discriminative power of the system was assessed by processing all pairs of model-object and test images, and counting the resulting amount of region matches. The highest ROC curve in figure i1 depicts the detection rate versus false-positive rate, while varying the detection threshold from 0 to 200 matches. The method performs very well, and can achieve 98% detection with 6% false-positives. For comparison, we processed the dataset also with 4 state-of-the-art affine region extractors [7,5,11,2], and described the regions with the SIFT [8] descriptor ², which has recently been demonstrated to perform best [12]. The matching is carried out by the 'unambiguous nearest-neighbor' approach ³ advocated in [11,8]: a model region is matched to the region of the test image with the closest descriptor if it is closer than 0.7 times the distance to the second-closest descriptor (the threshold 0.7 has been empirically determined to optimize results). Each of the central curves in i1 illustrates the behavior of a different extractor. As can be seen, none is satisfactory, which demonstrates the higher level of challenge offered by the dataset and therefore suggests that our approach can broaden the range of solvable OR cases. Closer inspection reveals the source of failure: typically only very few, if any, correct matches are produced when the object is present, which in turn is due to the lack of repeatability and the inadequacy of a simple matcher under such difficult conditions. The important improvement brought by the proposed method is best quantified by the difference between the highest curve and the central thick curve, representing the system we started from [2] (labeled '[2] org' in the plot).

The experiments confirm the power of the presented approach in solving very challenging cases. Moreover, non-rigid deformations are explicitly taken into account, and the approximate boundaries of the object is found, two features lacking in competing approaches [4,8,2,7,11,5,14]. The method is of general applicability, as it works with any affine invariant feature extractor. Future work aims at better exploiting the relationships between multiple model-views, at extending the scope to less richly textured objects, and at improving computational efficiency (currently, a 1.4 Ghz computer takes some minutes to process a pair of model and test images).

² All region extractors and the SIFT descriptor are implementations of the respective authors. We are grateful to Jiri Matas, Krystian Mikolajczyk, Andrew Zisserman, Cordelia Schmid and David Lowe for providing the programs.

³ We have also tried the standard approach, used in [7,5,2,12], which simply matches two nearest-neighbors if their distance is below a threshold, but it produced slightly worse results.

References

1. V. Ferrari, T. Tuytelaars and L. Van Gool, Wide-baseline Multiple-view Correspondences *IEEE Comp. Vis. and Patt. Rec.*, vol I, pp. 718–725, 2003.
2. T. Tuytelaars and L. Van Gool Wide Baseline Stereo based on Local, Affinely invariant Regions *Brit. Mach. Vis. Conf.*, pp. 412–422, 2000.
3. P.H.S. Torr and D. W. Murray The development and comparison of robust methods for estimating the fundamental matrix *IJCV*, 24(3), pp. 271–300, 1997.
4. F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, 3D Object Modeling and Recognition Using Affine-Invariant Patches and Multi-View Spatial Constraints, *IEEE Comp. Vis. and Patt. Rec.*, vol II, pp. 272–277, 2003.
5. K.Mikolajczyk and C.Schmid, An affine invariant interest point detector *European Conf. on Comp. Vis.*, vol. 1, pp. 128–142, 2002.
6. C.Schmid, Combining greyvalue invariants with local constraints for object recognition *IEEE Comp. Vis. and Patt. Rec.*, pp. 872–877, 1996.
7. S. Obdzalek and J. Matas, Object Recognition using Local Affine Frames on Distinguished Regions *Brit. Mach. Vis. Conf.*, pp. 414–431, 2002.
8. D. Lowe, Distinctive Image Features from Scale-Invariant Keypoints *submitted to Intl. Journ. of Comp. Vis.*, 2004
9. C. Cyr, B. Kimia, 3D Object Recognition Using Similarity-Based Aspect Graph *Intl. Conf. on Comp. Vis.*, 2001
10. H. Murase, S. Nayar, Visual Learning and Recognition of 3D Objects from Appearance *Intl. Journ. of Comp. Vis.*, 14(1), 1995
11. A. Baumberg, Reliable feature matching across widely separated views *IEEE Comp. Vis. and Patt. Rec.*, pp. 774–781, 2000
12. K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors *IEEE Comp. Vis. and Patt. Rec.*, vol II, pp. 257–263, 2003
13. F. Schaffalitzky, A. Zisserman Multi-view matching for unordered image sets *European Conf. on Comp. Vis.*, pp. 414–431, 2002.
14. F. Schaffalitzky, A. Zisserman Automated Scene Matching in Movies *CIVR*, 2002.
15. D. Tell, S. Carlsson Combining Appearance and Topology for Wide Baseline Matching *European Conf. on Comp. Vis.*, pp. 68–81, 2002.

Recognition by Probabilistic Hypothesis Construction

Pierre Moreels, Michael Maire, and Pietro Perona

California Institute of Technology, Pasadena CA 91125, USA,
`pmoreels@vision.caltech.edu`

Abstract. We present a probabilistic framework for recognizing objects in images of cluttered scenes. Hundreds of objects may be considered and searched in parallel. Each object is learned from a single training image and modeled by the visual appearance of a set of features, and their position with respect to a common reference frame. The recognition process computes identity and position of objects in the scene by finding the best interpretation of the scene in terms of learned objects. Features detected in an input image are either paired with database features, or marked as clutters. Each hypothesis is scored using a generative model of the image which is defined using the learned objects and a model for clutter. While the space of possible hypotheses is enormously large, one may find the best hypothesis efficiently – we explore some heuristics to do so. Our algorithm compares favorably with state-of-the-art recognition systems.

1 Introduction

In the computer vision literature there is broad agreement that objects and object categories should be represented as collections of parts (or features) which appear in a given mutual position or shape (eg side-by-side eyes, a nose below them etc). Each feature contains local information describing the image content [2,3]. There is, however, disagreement as to the best tradeoff in this design space. On one hand, one may wish to represent the appearance and position of parts in a careful probabilistic framework, which allows to generate principled learning and detection algorithms. One example of this approach is the ‘constellation model’ [4] which has been successfully applied to unsupervised learning and recognition of object categories amongst clutter [5,6]. This approach is penalized by a large number of parameters that are needed to represent appearance and shape and by algorithmic complexity – as a result there is a practical limit to the size of the models that one can use, typically limiting the number of object parts below 10. On the other hand, one finds in the literature models containing hundreds of features. In this case the authors dramatically simplify the way appearance and position are modeled as well as the algorithms used to learn and match models to images. A representative of this approach is David Lowe’s algorithm [7,8] which can recognize simultaneously and quickly multiple individual objects (as opposed to categories).

We are interested in exploring whether probabilistically rigorous modeling may be extended to yield practical data structures and algorithms for models that contain hundreds of features. To this end, we modify the constellation model [6,9] to incorporate a number of attractive features presented by Lowe: using a KD-tree for efficiently associating features by appearance as well as computing feature positions with respect to a common reference frame rather than with respect to each other. Additionally, we pool representational parameters amongst features. As a result, it is possible to learn models quickly based on a single example; additionally, the system gains the robustness associated with using a large number of features while also offering an expressive probabilistic model for verifying object presence. One additional contribution is exploring efficient algorithms for associating models with images that are based on this probabilistic model and the A* search technique [10,11].

In section 2 we review the feature matching and constellation model approaches upon which this paper builds. Section 3 details the probabilistic framework used in our recognition system. Section 4 describes the algorithm for incrementally constructing a high probability hypothesis without exploring the entire hypothesis space. In section 5, we discuss the task of learning. In section 6, we compare our system's performance against that of a pure feature matching approach. Finally, in section 7, we present conclusions and discuss areas for further research.

2 Related Research

A feature-based recognition approach recently developed by Lowe [7,8] consists of four stages: feature detection, extraction of feature correspondences, pose parameter estimation, and verification. Features are computed over multiple scales, at positions that are extrema of a difference-of-Gaussian function. An orientation is assigned to a feature using the histogram of local image gradients. Each feature's appearance is represented by a vector constructed from the local image region, sampled relative to the feature orientation. A k-d tree structure, modified with backtracking for search efficiency [12], is the central component of a database used to perform efficient appearance-based feature matching. Each match between scene and model features suggests a position, orientation, and scale for the model within the scene. Recognition is achieved by grouping similar model poses using a Hough transform and then explicitly solving for the transformation from model to scene coordinates.

The constellation model [4,5,6,9] also relies on matching image parts, but typically uses on the order of 5 features, whereas Lowe uses hundreds of features. Rather than restricting features to a rigid position, the constellation model uses a joint probability density on part positions. In addition, a probabilistic model for feature appearance is used, permitting the quality of matches to be measured. One drawback of the constellation model is the high number of training samples required, although recent work by Fei Fei et al [13] proved that learning can be efficiently achieved with few examples. Another disadvantage of the constellation

model lies in the large computation time required in order to learn feature configurations, limiting it to use of a relatively small number of parts for each object category. In our adaptation of the constellation approach for individual object recognition, this limitation disappears, at a slight cost to the model's generality.

3 Probabilistic Framework

We model individual objects as constellations of features. Features are generated by applying Lowe's feature detector [7,8] to each training image. Each feature has a position, orientation, and scale within the object model as well as a feature vector describing its appearance. We learn probabilistic models for foreground and background feature appearance. The collection of models extracted from training images, together with a k-d tree of model features searchable by appearance, forms the *database*.

Features are generated from a scene using the same procedure applied to training images. If a model is present, each of its features has a chance of appearing as a scene feature. We also expect spurious background detections in the scene. A *hypothesis* assigns each scene feature to either the background or a model feature. It also specifies the pose of each model present in the scene. A hypothesis may indicate the presence of multiple instances of the same object, each in a different pose.

The task of the recognition algorithm is to find the hypothesis that best explains the scene. The solution is the hypothesis with maximum probability conditioned on both the observed scene features and the database.

3.1 Hypothesis Valuation

Let \mathcal{O} denote the set of observed scene features, \mathcal{D} the database, and H a hypothesis. We define the valuation of H by $v(H) = p(H|\mathcal{O}, \mathcal{D})$. Using Bayes rule,

$$v(H) = p(H|\mathcal{O}, \mathcal{D}) = \frac{p(\mathcal{O}|H, \mathcal{D})p(H|\mathcal{D})}{p(\mathcal{O}|\mathcal{D})} \quad (1)$$

The desired output of the recognition algorithm is the hypothesis H maximizing this valuation. In particular,

$$H = \underset{H \in \mathcal{H}}{\operatorname{argmax}} \left(\frac{p(\mathcal{O}|H, \mathcal{D})p(H|\mathcal{D})}{p(\mathcal{O}|\mathcal{D})} \right) = \underset{H \in \mathcal{H}}{\operatorname{argmax}} (p(\mathcal{O}|H, \mathcal{D})p(H|\mathcal{D})) \quad (2)$$

where \mathcal{H} denotes the set of all hypotheses and we dropped the constant $p(\mathcal{O}|\mathcal{D})$.

In order to evaluate these probabilities, we expand a hypothesis into several components. The hypothesis states which objects are in the scene and where those objects are detected in the scene. Let m denote the number of object detections predicted by hypothesis H . Then, for $i = 1 \dots m$, H specifies the model, $M_i \in \mathcal{D}$, of the i^{th} detected object, as well a set of parameters, Z_i , describing that model's pose in the scene.

In addition to stating the position of detected objects, hypothesis H attributes their appearance to features found in the scene. In particular, H breaks the set of scene features, \mathcal{O} , into $m + 1$ disjoint sets, $\mathcal{O}_0 \dots \mathcal{O}_m$, where \mathcal{O}_0 is the set of features attributed to the background and for $i = 1 \dots m$, \mathcal{O}_i is the set of features attributed to model M_i . To specify the exact pairing between scene features in \mathcal{O}_i and model features in M_i , we introduce two auxiliary variables, \mathbf{d}_i and \mathbf{h}_i . The binary vector \mathbf{d}_i indicates which features of M_i are detected (value 1) and which features are missing (value 0). Vector \mathbf{h}_i also contains an entry for each feature j of M_i . If j is detected ($\mathbf{d}_{ij} = 1$), then \mathbf{h}_{ij} indicates the element of \mathcal{O}_i to which j corresponds. In other words, \mathbf{h}_i maps indices of detected model features to indices of their corresponding scene features.

For notational convenience, we define the single vector \mathbf{h} to contain the entire correspondence map between scene features and model features (or background). \mathbf{h} is simply the concatenation of all the \mathbf{h}_i 's. Also, n denotes the number of background features, or equivalently, the size of \mathcal{O}_0 . Together, \mathbf{h} , n , $\{\mathbf{d}_1, \dots, \mathbf{d}_m\}$, $\{Z_1, \dots, Z_m\}$, and $\{M_1, \dots, M_m\}$ completely specify a hypothesis. These variables contain all detection, pose, and feature correspondence information.

Using this decomposition, we now return to the computation of the valuation of a hypothesis. From equation (2) we can redefine the hypothesis valuation as

$$v'(H) = p(\mathcal{O}|H, \mathcal{D}) \cdot p(H|\mathcal{D}) \quad (3)$$

3.2 Pose and Appearance Density

The term $p(\mathcal{O}|H, \mathcal{D})$ characterizes the probability density in location, scale, orientation, and appearance for the features detected in the scene image. Conditioning on the pose of models present in hypothesis H , we can assume that features attributed by H to different model objects are mutually independent:

$$\begin{aligned} p(\mathcal{O}|H, \mathcal{D}) &= p(\mathcal{O}|\mathbf{h}, n, \{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}, \mathcal{D}) \\ &= p(\mathcal{O}_0|n, \mathcal{D}) \cdot \prod_{i=1}^m p(\mathcal{O}_i|\mathbf{h}_i, \mathbf{d}_i, Z_i, M_i, \mathcal{D}) \end{aligned} \quad (4)$$

- $p(\mathcal{O}_0|n, \mathcal{D})$ is the probability that the n background detections would occur at the exact positions and with the exact appearances specified in \mathcal{O}_0 . We assume each point in the (location, orientation, scale) space examined by the feature generator has an equal chance of producing a spurious detection. Assuming that clutter detections are independent from each other,

$$p(\mathcal{O}_0|n, \mathcal{D}) = \left[\frac{1}{A} \cdot \frac{1}{2\pi} \right]^n \cdot \prod_{\mathbf{x} \in \mathcal{O}_0} p_{\mathbf{bg}}(\mathbf{x}|\mathcal{D}) \quad (5)$$

where A is the number of pixels in the Gaussian pyramid used for feature detection, or equivalently, the size of the (location, scale) space, and there is a range of 2π in possible values for orientation. $p_{\mathbf{bg}}(\mathbf{x}|\mathcal{D})$ is the density describing the appearance of background features.

- $p(\mathcal{O}_i|\mathbf{h}_i, \mathbf{d}_i, Z_i, M_i, \mathcal{D})$ is the probability that the detections of the model features indicated by the hypothesis would occur with the exact pose and appearance specified in \mathcal{O}_i . Conditioning on model pose, we will assume independence between model features. This is a key assumption distinguishing our model from the constellation model [4,5,6]. Thus,

$$p(\mathcal{O}_i|\mathbf{h}_i, \mathbf{d}_i, Z_i, M_i, \mathcal{D}) = \prod_{\mathbf{x} \in \mathcal{O}_i} p_{\text{pose}}(\mathbf{x}|\mathbf{h}_i, \mathbf{d}_i, Z_i, M_i, \mathcal{D}) \cdot p_{\text{fg}}(\mathbf{x}|\mathbf{h}_i, \mathbf{d}_i, M_i, \mathcal{D}) \quad (6)$$

where p_{pose} and p_{fg} are the pose and appearance probabilities, respectively, for the foreground features.

The discussion on learning in section 5 describes the technique used for estimating probability densities p_{bg} , p_{pose} , and p_{fg} .

3.3 Hypothesis Prior

The term $p(H|\mathcal{D})$ is the prior on the hypothesis. We expand this term as

$$\begin{aligned} p(H|\mathcal{D}) &= p(\mathbf{h}, n, \{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}|\mathcal{D}) \\ &= p(\mathbf{h}|n, \{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}, \mathcal{D}) \cdot p(n|\{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}, \mathcal{D}) \\ &\quad \cdot \left[\prod_{i=1}^m p(\mathbf{d}_i|Z_i, M_i, \mathcal{D}) \right] \cdot p(\{Z_i\}, \{M_i\}|\mathcal{D}) \end{aligned} \quad (7)$$

- $p(\mathbf{h}|n, \{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}, \mathcal{D})$ is the probability of a specific set of feature assignments. As \mathbf{h} is simply a vector of indices mapping model features to scene features, and we have no information on scene feature appearance or position at this stage, all mappings that predict n background features and are consistent with the detection vectors $\{\mathbf{d}_i\}$ are equally likely. Hence,

$$p(\mathbf{h}|n, \{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}, \mathcal{D}) = p(\mathbf{h}|n, \{\mathbf{d}_i\}) = \begin{cases} \left[\frac{N!}{(N-N_{fg})!} \right]^{-1} & \mathbf{h}, n, \{\mathbf{d}_i\} \\ & \text{consistent} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where N is the total number of features in the scene image and $N_{fg} = N - n$ is the number of foreground features predicted by the hypothesis.

- $p(n|\{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}, \mathcal{D})$ is the probability of obtaining n background features. Background features are spurious responses to the feature detector that do not match with any known object. We assume a Poisson distribution for the number of background features [9]. Since scene images may have different sizes, the expected number of background detections is proportional to the area A examined by the feature detector. If λ denotes the mean number of background features per unit area, then

$$p(n|\{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}, \mathcal{D}) = p_{\text{Poisson}}(n|\lambda, A) = e^{-\lambda A} \frac{(\lambda A)^n}{n!} \quad (9)$$

- $p(\mathbf{d}_i|Z_i, M_i, \mathcal{D})$ is the probability of detecting the indicated model features from model M_i . Let p_{ij} denote the probability that feature j of model M_i is detected in the scene. The probability that it is missing is $(1 - p_{ij})$. We break $p(\mathbf{d}_i|Z_i, M_i, \mathcal{D})$ into a term for detected features and a term for missing features to obtain

$$p(\mathbf{d}_i|Z_i, M_i, \mathcal{D}) = \prod_{\substack{j \text{ detected} \\ (\mathbf{d}_{ij}=1)}} p_{ij} \cdot \prod_{\substack{j \text{ missing} \\ (\mathbf{d}_{ij}=0)}} (1 - p_{ij}) \quad (10)$$

- $p(\{Z_i\}, \{M_i\}|\mathcal{D})$ is the prior on detecting objects $\{M_i\}$ in poses $\{Z_i\}$. We model this prior by a uniform density over frame transformations and combinations of model objects in the scene. Thus, this term is dropped in the implementation presented here.

4 Hypothesis Search

The recognition process consists of finding the hypothesis H that maximizes $v'(H)$. Unfortunately, due to the size of the hypothesis space \mathcal{H} , it is not possible to evaluate $v'(H)$ for each $H \in \mathcal{H}$. Early work by Grimson (e.g. [14]) showed the exponential growth of the search tree and the need for hypotheses pruning. Here, we use the A* search technique [10,11] to incrementally construct a reasonable hypothesis while only examining a small fraction of the hypothesis space.

In constructing incrementally a solution, we introduce the notion of a *partial hypothesis* to refer to a partial specification of a hypothesis. In particular, a partial hypothesis specifies a set of models $\{M_i\}$ and their corresponding poses $\{Z_i\}$ as well as a pairing between scene features and model features. Unpaired scene features are either marked as background or unassigned, whereas are either missing or unassigned. The partial hypothesis does not dictate how the unassigned scene or model features are to be treated. A *completion* of a partial hypothesis is a hypothesis that makes the same assignments as the partial hypothesis, but in which there are no unassigned scene features. A completion may introduce new models, make pairings between unassigned scene and model features, mark unassigned scene features as background, or mark unassigned model features as missing.

4.1 A*

We can organize the set of all partial hypotheses into a tree. The root of the tree is the partial hypothesis containing no models and in which all scene features are unassigned. The leaves of the tree are all complete hypotheses, (i.e. \mathcal{H}). Descending a branch of the tree corresponds to incrementally making decisions about feature assignments in order to further specify a partial hypothesis.

We prioritize the exploration of the tree by computing a valuation for each partial hypothesis. Partial hypotheses are entered into a priority queue according to this valuation. At each step of the search procedure, the highest valuation

partial hypothesis is dequeued and split into two new partial hypotheses. In one of these new hypotheses, a certain feature assignment is made. In the other new hypothesis, that feature assignment is expressly forbidden from occurring. This binary splitting ensures that a search of the hypothesis tree visits each partial hypothesis at most once.

4.2 Partial Hypothesis Valuation

To produce an effective search strategy, the valuation of a partial hypothesis should reflect the valuation of its best possible completion. If these two quantities were equal, the search would immediately descend the tree to the best complete hypothesis. However, it is impossible to compute the valuation of the best possible completion before actually finding this completion, which is the task of the search in the first place. Therefore, we will define the valuation of a partial hypothesis using a heuristic.

The heuristic we use can be thought of as the “optimistic worst-case scenario”. It is the valuation of the partial hypothesis’s completion in which all unassigned scene features are marked as background and all unassigned model features are dropped from the model. Unassigned model features are counted as neither detected nor missing. They do not enter into probability computations.

Note that this choice of heuristic is coherent with the expression for the valuation of a complete hypothesis. As the algorithm makes assignments in a partial hypothesis, its valuation approaches the valuations of its possible completions. Furthermore, this valuation is likely to serve as a decent guide for the search procedure. It is a measure of the minimum performance offered by a branch under the assumption that further assignments along that branch will do no harm.

4.3 Initialization

A list of potential database feature matches is created for each scene feature based on appearance. The empty partial hypothesis is split into two based on the best appearance match. One subbranch accepts this match, the other rejects it and forbids it.

4.4 Search Step

The partial hypothesis H with the highest valuation is dequeued. If H contains a model in which there are unassigned features, the algorithm picks one of these unassigned model features. A similar splitting to that in the initialization step is performed: one subbranch adds the match to the hypothesis, and the other forbids it as far as this hypothesis is concerned. In order to save computation time, we greedily follow only the branch that results in a better valuation. This is reasonable for rigid models in which the pose constraints should allow very few possibilities for a correct match in the scene.

If there are no unassigned model features in H , we pick the unassigned scene feature with the best appearance based match and split the hypothesis on this

assignment: one subbranch accepts the match and adds the corresponding model to the hypothesis, the other rejects the match, and adds no information regarding this model.

In both of the above cases, the resulting partial hypothesis or hypotheses are enqueued and the process is repeated.

4.5 Termination

The search process corresponding to one object terminates when no more unassigned features are available in this object. The scene features paired with this object are removed, and the search iterates with the remaining scene features. If all model objects have been considered without fully explaining the scene, the unassigned scene features are considered as background detections.

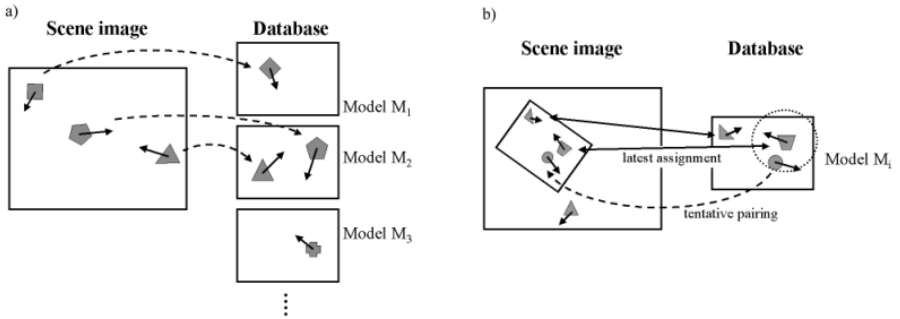


Fig. 1. Sketch of hypothesis build: a) Initialization: The best appearance match in the database is identified for each scene feature. Each such match is entered in the queue as a partial hypothesis. b) Search for a new match in the partial hypothesis which has highest valuation: we look for an unassigned feature in the same model image M_i . This feature is mapped to its best appearance match in the scene, if this new pairing is coherent with the pose predicted by the hypothesis - otherwise, the match is rejected. The pose is then updated based on the new match.

5 Learning

Several components of the probabilistic framework given above must be inferred from training examples. Since our system requires only a single training image per object, we cannot estimate separate appearance and pose densities for each feature in an object model. We therefore utilize the entire feature database in estimating global probability densities which can be applied to all features. Note that only training images are used here, not the test set.

5.1 Background Features

To estimate the background appearance density, we assume that a typical background feature looks like a feature found in the database. The background, like the database, is composed of objects. It just happens that these objects are not in the database. A probability density for the appearance of features in the database describes the appearance of background detections in a scene. We model this density with a full covariance gaussian density. Letting μ_{bg} and Σ_{bg} denote the mean and covariance of the database feature appearance vectors,

$$p_{\text{bg}}(\mathbf{x}|\mathcal{D}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_{\text{bg}}|} e^{-\frac{1}{2}(\mathbf{x}_{\text{app}} - \mu_{\text{bg}})^T \Sigma_{\text{bg}}^{-1} (\mathbf{x}_{\text{app}} - \mu_{\text{bg}})} \quad (11)$$

where \mathbf{x}_{app} is the appearance vector of feature \mathbf{x} and d the dimension of appearance vectors.

A typical model generates 500 to 1000 features, resulting for a database with 100 objects in a total of 50,000 to 100,000 training examples for the background appearance density. As our experiments used 128-dimensional appearance vectors, this was a sufficient number of examples for estimating the gaussian density.

The mean number of background detections per unit area, λ , is programmer specified in our current implementation. When running Lowe's detection method on our training and test sets, 80% of the detections were assigned to the background, therefore we chose this same fraction for λ . This parameter has only weakly effects on the total probability as the terms for pose and appearance dominate.

5.2 Foreground Features

The foreground appearance density must describe how closely a scene feature resembles the model feature to which it is matched. This density is difficult to estimate as in principle, it involves establishing hundreds of thousands of ground truth matches by hand. A possible shortcut is looking at statistics coming from planar scenes seen from different viewpoints [15], or synthetic deformations of an image [3].

Here we followed a different approach: we approximate a good match for a feature by its closest match in appearance in the database. The difference in appearance between correctly matched foreground features is modeled with a gaussian density with full covariance matrix, and the covariance matrix Σ_{fg} is estimated from the difference in appearance between database features paired in such a manner. This yields

$$p_{\text{fg}}(\mathbf{x}|\mathbf{h}_i, \mathbf{d}_i, M_i, \mathcal{D}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_{\text{fg}}|} e^{-\frac{1}{2}(\mathbf{x}_{\text{app}} - \mathbf{y}_{\text{app}})^T \Sigma_{\text{fg}}^{-1} (\mathbf{x}_{\text{app}} - \mathbf{y}_{\text{app}})} \quad (12)$$

where $\mathbf{y} = \mathbf{h}_i^{-1}(\mathbf{x})$ is the model feature paired with scene feature \mathbf{x} .

Unlike background feature pose which are modeled with a uniform distribution in equation (5), foreground features are expected to lie in a pose consistent

with that of their corresponding model. In particular, model pose Z_i predicts a scene location, orientation, and scale for each feature of M_i . If the hypothesis matches scene feature \mathbf{x} to model feature \mathbf{y} , and Z_i maps \mathbf{y} to \mathbf{z} , we write

$$p_{\text{pose}}(\mathbf{x}|\mathbf{h}_i, \mathbf{d}_i, Z_i, M_i, \mathcal{D}) = G_{\text{loc}}(\mathbf{x}|\mathbf{z}) \cdot G_{\theta}(\mathbf{x}|\mathbf{z}) \cdot G_{\text{s}}(\mathbf{x}|\mathbf{z}) \cdot \quad (13)$$

where G_{loc} , G_{θ} , and G_{s} are Gaussian densities for location, orientation, and log scale, respectively, with means given by the pose of \mathbf{z} . The covariance parameters of these densities are currently specified by hand, with values of 20 pixels for location, half an octave for log-scale and 60 degrees for orientation (orientation was quite unreliable).

We determine the model pose Z_i by solving for the similarity transform that minimizes, in the least-squares sense, the distance between observed and predicted locations of foreground model features. Z_i is updated whenever a previously unassigned feature of M_i is matched.

The probability p_{ij} of detecting individual features is set to the same value across features and models. A reasonable choice is the fraction of features that are typically needed to produce a reliable pose estimate. This value was obtained by running Lowe’s detection method on our training and test sets: in average 20% of a model features were found in a test image containing this model. This value of 20% was used for p_{ij} .

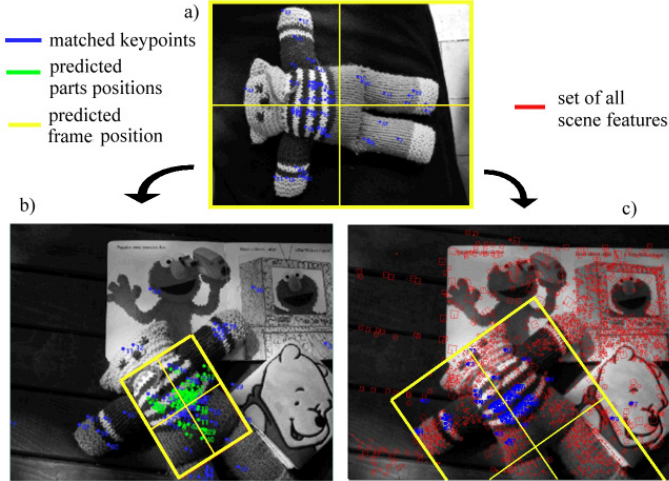


Fig. 2. Example of result for a textured object included in a complex scene (only one detection shown here). According to this hypothesis, the box displayed in the model image is transformed into the box shown in the scene image. a) Initial object b) Result of Lowe’s algorithm. Since the stuffed bear is a textured object, detection of similar features can occur in many locations, leading to incorrect pairings. As a result, the frame transformation, estimated only from the features positions, is inaccurate. c) Result of the probabilistic search.

6 Experimental Results

In the absence of “standard” training and test sets of images containing both objects and clutter, we compare the performance of our probabilistic search method to that of Lowe’s algorithm on a training set consisting of 100 images of toys and common kitchen items, with a single image per object. The test set contained images of single objects, as well as complicated scenes that include several objects, ranging from 1 to 9 objects. It included 80 test images, with a total of 254 objects to be detected (each object was considered as one detection). Some test images didn’t contain any learned object. In that case all feature detections are expected to be assigned to the background. We used a resolution of 480×320 for training images and test images of single objects, and 800×533 for complex scenes. All images were taken in a kitchen, with an off-the-shelf digital camera, and no precautions were taken with respect to lighting conditions, viewpoint angle and background. In particular, the lighting conditions varied significantly between training and test images, and viewing angles varied between 0 and 180 degrees (picture of the back of an object taken as test while the corresponding model was a picture of the front of the object). No image was manually segmented, and the proportion of features generated by an object in a model or test image, ranged from 10% to 80% (80% for a single object occupying most of the image). The database is available online from <http://www.vision.caltech.edu/html-files/archive.html>.

Our algorithm achieved a detection performance similar to Lowe’s system, with a detection rate of 85%. Figure 3 shows ROC curves for both methods. The threshold used is the accuracy of the best hypothesis at the end of the search. Since our method verifies the coherence of each match by scoring partial hypothesis, our false alarm rate was lower than that of Lowe’s method.

In order to measure the accuracy of the pose transformations estimated by each method, the training and test images were manually marked with ground truth information. An ellipse was fitted, and a canonical orientation was chosen, for each object. We measured the accuracy of the transformation with the distance in pixels, between the predicted positions of the ellipses in a scene, and the ground truth previously recorded. The error was averaged across points regularly spaced on the ellipse and across test images. We obtained a mean error of 45 pixels for our method, and 56 pixels for Lowe’s algorithm.

Our approach requires to examine and evaluate a number of partial and complete hypotheses that is much higher than with Lowe’s method. As a result, the probabilistic algorithm is the slower of the two methods. Our unoptimized code for Lowe’s method takes in average 2 seconds on a Pentium 4 running at 2.4GHz to identify objects in a 800×533 image, while our probabilistic algorithm requires on average 10 seconds for the same image.

In practice, the A* search achieves only little pruning, typically 10-20% of the branches are eliminated. Therefore, the valuation heuristic was coupled with a stopping criterion (depth-first completion of the partial hypothesis that performs best after 4000 iterations). The main computational benefit of the A* method in this paper, is to introduce a framework for evaluating partial hypotheses in a

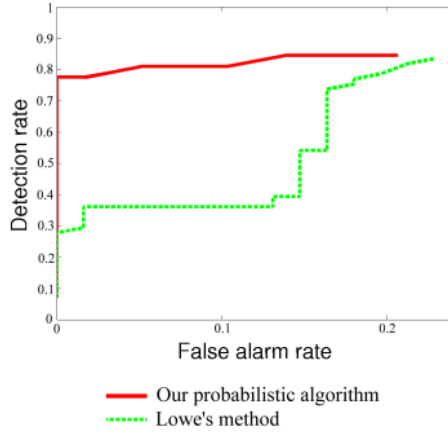


Fig. 3. ROC based on the accuracy of the pose estimated by the best hypothesis. It measures how much the hypothesis' prediction of the object position, differs from the ground truth. This quantity can be measured for both recognition systems.

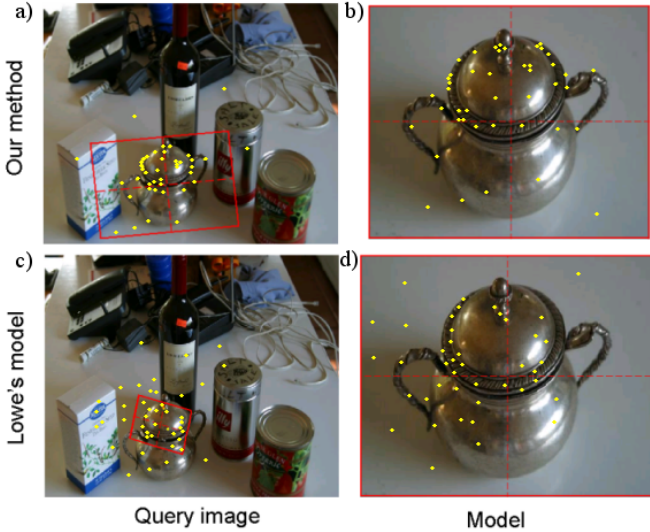


Fig. 4. Other example of recognition in a complex environment. a) and b) present one match obtained by our probabilistic search, c) and d) are the best result from Lowe's voting approach. Since Lowe's method does not evaluate geometric and appearance quality of hypotheses, numerous incorrect correspondences are accepted. As a result, the estimated frame position is inaccurate. The probabilistic search accepts only matches that are geometrically coherent, and leads to accurate pose parameters.



Fig. 5. Samples from our training and test sets. The red boxes show locations where models were identified

way that is coherent with the valuation of complete hypotheses, and a ranking of hypotheses that leads to efficient search.

7 Discussion and Conclusion

We have presented a new probabilistic model and efficient search strategy for recognizing multiple objects in images. Our model provides a unified view of two previous lines of research: it may be thought as a probabilistic interpretation of David Lowe’s work [7,8] or, conversely, as a special case of the constellation model [4] where many of the parameters are pooled amongst models, rather than learned individually.

Our experiments indicate that the system we propose achieves the same detection rate as Lowe’s algorithm with significantly lower false alarm rates. The localization error of detected objects is also smaller. The price to be paid is a slower processing time, although this may not be a significant issue since our code is currently not optimized for speed. The front-end of both systems was identical (feature detection, feature representation, feature matching) and therefore all measurable differences are to be ascribed to the probabilistic model and to the matching algorithm.

It is clear that the heuristic we chose for ranking partial hypotheses is susceptible of improvement. In choosing it we followed intuition rather than a principled

approach. This is obviously an area for further investigation. Developing better techniques for estimating the probability density function of appearance and pose error of both foreground and background features is another issue deserving of further attention.

Acknowledgments. The authors thank Prof. D.Lowe for kindly providing the code of his feature detector and for useful advice. They also acknowledge funding from the NSF Engineering Research Center on Neuromorphic Systems Engineering at Caltech.

References

1. Clarke, F., Ekeland, I.: Nonlinear oscillations and boundary-value problems for Hamiltonian systems. *Arch. Rat. Mech. Anal.* **78** (1982) 315–333
2. B. Schiele: Object Recognition Using Multidimensional Receptive Field Histograms PhD thesis, I.N.P. de Grenoble, 1997.
3. C. Schmid and R. Mohr: Local greyvalue invariants for image retrieval *IEEE Trans. on Patt. Anal. Mach. Int.*, 19(5):530–535, 1997.
4. M.C. Burl and P. Perona Recognition of planar object classes *IEEE Comp. on Comp. Vision and Patt. Recog.*, CVPR 96, San Francisco, CA, June 1996.
5. R. Fergus, P. Perona, A. Zisserman: Object class recognition by unsupervised scale-invariant learning. *IEEE Conf. on Comp. Vision and Patt. Recog.*, 2003
6. M. Weber, M. Welling and P. Perona: Unsupervised learning of models for recognition. *Proo. 6th Europ. Conf. Comp. Vis.*, ECCV2000, 2000.
7. D.G. Lowe: Object recognition from local scale-invariant features. *Proc. Int. Conf. Comp. Vision*, Corfu, Greece, pp. 1150–1157, 1999.
8. D.G. Lowe: Distinctive image features from scale-invariant keypoints accepted paper, *Int. J. of Comp. Vision*, 2004.
9. M. Weber: Unsupervised Learning of Models for Object Recognition, Ph.D thesis, Department of Computation and Neural Systems, California Institute of Technology, Pasadena, CA, 2000.
10. J.M. Coughlan and A.L.Yuille: Bayesian A* tree search with expected $O(N)$ node expansions: applications to road tracking. Draft submitted to *Neural Computation*, Dec. 2002.
11. J. Pearl: *Heuristics*, Addison-Wesley, 1984.
12. J.S. Beis and D.G. Lowe: Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, Puerto Rico, pp. 1000-1006, 1997.
13. L. Fei-Fei, R. Fergus and P. Perona: A bayesian approach to unsupervised one-shot learning of object categories, *Proc. Int. Conf. on Comp. Vision*, Nice, France, 2003.
14. W.E.L. Grimson: Model-based recognition and localization from sparse range or tactile data, *AI Memo 738*, Massachusetts Institute of Technology, Aug 1983.
15. K. Mikolajczyk and C. Schmid, A performance evaluation of local descriptors, *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, Madison, Wisconsin, p. 257.

Human Detection Based on a Probabilistic Assembly of Robust Part Detectors

K. Mikolajczyk¹, C. Schmid², and A. Zisserman¹

¹ Dept. of Engineering Science
Oxford, OX1 3PJ, United Kingdom
{km,az}@robots.ox.ac.uk

² INRIA Rhône-Alpes
38330 Montbonnot, France
schmid@inrialpes.fr

Abstract. We describe a novel method for human detection in single images which can detect full bodies as well as close-up views in the presence of clutter and occlusion. Humans are modeled as flexible assemblies of parts, and robust part detection is the key to the approach. The parts are represented by co-occurrences of local features which captures the spatial layout of the part's appearance. Feature selection and the part detectors are learnt from training images using AdaBoost.

The detection algorithm is very efficient as (i) all part detectors use the same initial features, (ii) a coarse-to-fine cascade approach is used for part detection, (iii) a part assembly strategy reduces the number of spurious detections and the search space. The results outperform existing human detectors.

1 Introduction

Human detection is important for a wide range of applications, such as video surveillance and content-based image and video processing. It is a challenging task due to the various appearances that a human body can have. In a general context, as for example in feature films, people occur in a great variety of activities, scales, viewpoints and illuminations. We cannot rely on simplifying assumptions such as non-occlusion or similar pose. Of course, for certain applications, such as pedestrian detection, some simplifying assumptions lead to much better results, and in this case reliable detection algorithms exist. For example, SVM classifiers have been learnt for entire pedestrians [14] and also for rigidly connected assemblies of sub-images [13]. Matching shape templates with the Chamfer distance has also been successfully used for pedestrian detection [1,5]. There is a healthy line of research that has developed human detectors based on an assembly of body parts. Forsyth and Fleck [4] introduced body plans for finding people in general configurations. Ioffe and Forsyth [6] then assembled body parts with projected classifiers or sampling. However, [4,6] rely on simplistic body part detectors – the parts are modelled as bar-shaped segments and pairs of parallel edges are extracted. This body part detector fails in the presence

of clutter and loose clothing. Similarly, Felzenszwalb and Huttenlocher [2] show that dynamic programming can be used to group body plans efficiently, but simplistic colour-based part detectors are applied. An improvement on body part detection is given in Ronfard *et al.* [17] where SVMs are trained for each body part. An improvement on the modelling of body part relations is given in Sigal *et al.* [21], where these are represented by a conditional probability distribution. However, these relations are defined in 3D, and multiple simultaneous images are required for detection.

In this paper we present a robust approach to part detection and combine parts with a joint probabilistic body model. The parts include a larger local context [7] than in previous part-based work [4,17] and they therefore capture more characteristic features. They are however sufficiently local (cf. previous work on pedestrian detectors [14]) to allow for occlusion as well as for the detection of close-up views. We introduce new features which represent the shape better than the Haar wavelets [14], yet are simple enough to be efficiently computed. Our approach has been inspired by recent progress in feature extraction [10,18,19,20], learning classifiers [15,22] and joint probabilistic modelling [3].

Our contribution is three-fold. Firstly, we have developed a robust part detector. The detector is robust to partial occlusion due to the use of local features. The features are local orientations of gradient and Laplacian based filters. The spatial layout of the features, together with their probabilistic co-occurrence, captures the appearance of the part and its distinctiveness. Furthermore, the features with the highest occurrence and co-occurrence probabilities are learnt using AdaBoost. The resulting part detector gives face detection results comparable to state of the art detectors [8,22] and is sufficiently general to successfully deal with other body parts. Secondly, the human detection results are significantly improved by computing a likelihood score for the assembly of body parts. The score takes into account the appearance of the parts and their relative position. Thirdly, the approach is very efficient since (i) all part detectors use the same initial features, (ii) a coarse-to-fine cascade approach successively reduces the search space, (iii) an assembly strategy reduces the number of spurious detections.

The paper is structured as follows. We introduce the body model in section 2. We then present the robust part detector in section 3, and the detection algorithm in section 4. Experimental results are given in section 5.

2 Body Model

In this section we overview the body model which is a probabilistic assembly of a set of body parts. The joint likelihood model which assembles these parts is described in section 2.1. The body parts used in the model are given in section 2.2, and geometric relations between the parts in section 2.3.

2.1 Joint Likelihood Body Model

Our classification decision is based on two types of observations, which correspond to the body part appearance and relative positions of the parts. The appearance is represented by features F and the body part relations by geometric parameters \mathcal{R} . The form of a Bayesian decision for body B is:

$$\frac{p(B|\mathcal{R}, \mathcal{F})}{p(\text{non } B|\mathcal{R}, \mathcal{F})} = \frac{p(\mathcal{R}|\mathcal{F}, B)}{p(\mathcal{R}|\mathcal{F}, \text{non } B)} \cdot \frac{p(\mathcal{F}|B)}{p(\mathcal{F}|\text{non } B)} \cdot \frac{p(B)}{p(\text{non } B)} \quad (1)$$

The first term of this expression is the probability ratio that body parts are related by geometric parameters measured from the image. The second term is the probability ratio that the observed set of features \mathcal{F} belong to a body:

$$\frac{p(\mathcal{F}|B)}{p(\mathcal{F}|\text{non } B)} = \prod_{f \in \mathcal{F}} \frac{p(f, \mathbf{x}_f|B)}{p(f, \mathbf{x}_f|\text{non } B)}$$

This set consists of a number of local features f and their locations \mathbf{x}_f in a local coordinate system attached to the body. The third term of (1) is a prior probability of body and non-body occurrence in images. It is usually assumed constant and used to control the false alarm rate.

Individual body part detectors are based on appearance (features and their locations) and provide a set of candidates for body parts. This is discussed in section 3. Given a set of candidate parts the probability of the assembly (or a sub-assembly) is computed according to (1). For example, suppose that a head H is detected based on the appearance, i.e. $p(\mathcal{F}|H)/p(\mathcal{F}|\text{non } H)$ is above threshold, then the probability that an upper body (U) is present can be computed from the joint likelihood of the upper-body/head sub-assembly $p(U, H)$. Moreover, a joint likelihood can be computed for more than two parts. In this way we can build a body structure by starting with one part and adding the confidence provided by other body part detectors. Implementation details are given in section 4.

2.2 Body Parts

In the current implementation we use 7 different body parts as shown in Figure 1. There are separate parts for a frontal head (a bounding rectangle which includes

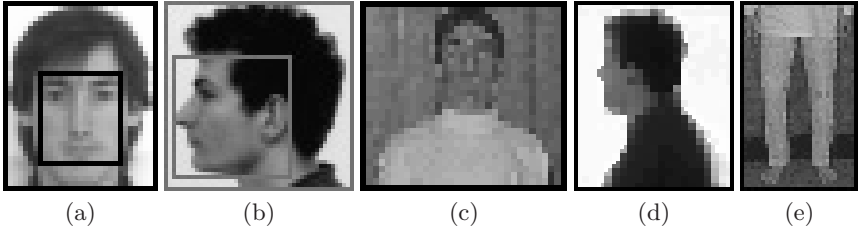


Fig. 1. Body parts. (a) Frontal head and face (inner frame). (b) Profile head and face (inner frame). (c) Frontal upper body. (d) Profile upper body. (e) Legs.

the hair), and face alone. Similarly there is a profile head part and a profile face part.

Each body part is detected separately, as described in section 3 based on its likelihood ratio.

2.3 Body Geometric Relations

The probability of a false positive for an individual detector is higher then for several detectors with a constraint on geometric relations between parts. The geometric relationship between the parts is here represented by a Gaussian $G(x_1 - x_2, y_1 - y_2, \sigma_1/\sigma_2)$ depending on their relative position and relative scale. σ_1 and σ_2 correspond to the scales (sizes) at which two body parts are detected. These parameters are learnt from training data. The size of a human head can vary with respect to the eyes/mouth distance. Similarly, the scale and the relative location between other body parts can vary for people. Figure 2(b) shows the Gaussian estimated for the head location with respect to the face location. Figure 2(c-d) shows the geometric relations for other body parts. We need to estimate only one Gaussian relation between two body parts, since the Gaussian function in the inverse direction can be obtained by appropriately inverting the parameters. Note that each of the detectors allows for some variation in pose. For example, the legs training data covers different possible appearance of the lower body part.

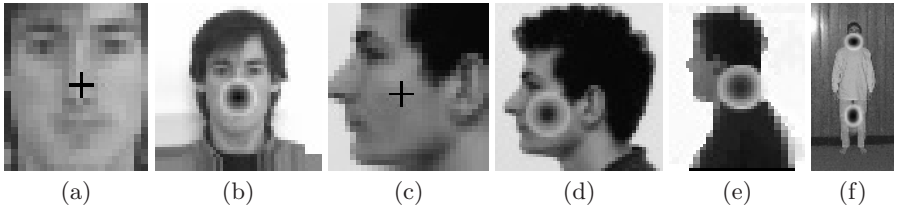


Fig. 2. Gaussian geometric relations between body parts. (a) Frontal face location. (b) Frontal head location with respect to the face location. (c) Profile location. (d) Profile head location with respect to the profile location. (e) Profile upper body location with respect to the head. (f) Frontal upper body location with respect to the head location, and legs with respect to the upper body.

3 Body Part Detector

In this section we present the detection approach for individual body parts. In sections 3.1 and 3.2 we describe the low-level features and the object representation. Section 3.3 explains the classifiers obtained from the features and the learning algorithm.

3.1 Orientation Features

An object's appearance is represented by orientation-based features and local groupings of these features. This choice is motivated by the excellent performance of SIFT descriptors [10,12] which are local histograms of gradient orientations. SIFT descriptors are robust to small translation and rotation, and this is built into our approach in a similar way.

Orientation features. Our features are the dominant orientation over a neighbourhood and are computed at different scales. Here we use 5 scale levels and a 3-by-3 neighbourhood. Orientation is either based on first or second derivatives.

In the case of first derivatives, we extract the gradient orientation. This orientation is quantized into 4 directions, corresponding to horizontal, vertical and two diagonal orientations. Note that we do not distinguish between positive and negative orientations. We then determine the score for each of the orientations using the gradient magnitude. The dominant direction is the one which obtains the best score. If the score is below a threshold, it is set to zero. Figure 3(b) shows the gradient image and Figure 3(c) displays the dominant gradient orientations where each of the 5 values is represented by a different gray-level value. Note the groups of dominant orientations on different parts of the objects.

A human face can be represented at a very coarse image resolution as a collection of dark blobs. An excellent blob detector is the Laplacian operator [9]. We use this filter to detect complementary features like blobs and ridges. We compute the Laplacian ($d_{xx} + d_{yy}$) and the orientation of the second derivatives ($\arctan(d_{yy}/d_{xx})$). We are interested in dark blobs therefore we discard the negative Laplacian responses, since they appear on bright blobs. Figure 3(d) shows the positive Laplacian responses. Similarly to the gradient features we select the dominant orientation. Second derivatives are symmetrical therefore their responses on ridges of different diagonal orientations are the same. Consequently there are 3 possible orientations represented by this feature. Figure 3(e) displays the dominant second derivative orientations where each orientation is represented by a different gray-level value.

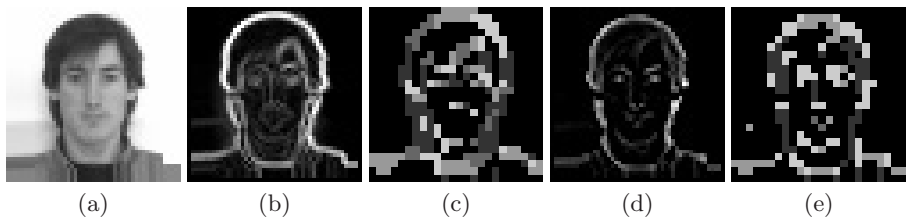


Fig. 3. Orientation features. (a) Head image. (b) Gradient image. (c) Dominant gradient orientations. (d) Positive Laplacian responses. (e) Dominant orientations of the second derivatives.

Feature groups. Since a single orientation has a small discriminatory power, we group neighbouring orientations into larger features. The technique described below was successfully applied to face detection [11,19]. We use two different combinations of local orientations. The first one combines 3 neighbouring orientations in a horizontal direction and the second one combines 3 orientations in a vertical direction. Figure 4(a) shows the triplets of orientations. A single integer value is assigned to each possible combination of 3 orientations. The number of possible values is therefore $v_{max} = 5^3 = 125$ for the gradient and $v_{max} = 4^3 = 64$ for the Laplacian. More than 3 orientations in a group significantly increase the number of possible combinations and poorly generalize. In summary, at a given scale there are four different feature group types v_t : horizontal and vertical groups for gradient orientations and horizontal and vertical groups for the Laplacian.

3.2 Object Representation

The location of a feature group on the object is very important as we expect a given orientation to appear more frequently at a particular location and less frequently at the other locations. The location is specified in a local coordinate system attached to the object (Figure 4(b)). To make the features robust to small shifts in location and to reduce the number of possible feature values we quantize the locations into a 5×5 grid (Figure 4(c)).

In the following we will use the notation (x, y, v_t) to refer to a feature group of type v_t at the grid location (x, y) . For simplicity we will refer to this as a *feature* (x, y, v_t) .

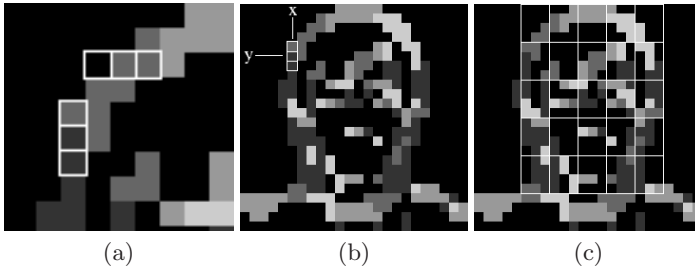


Fig. 4. Local groups of features. (a) Two groups of local orientations. (b) Location of the feature on the object. (c) Grid of quantized locations.

3.3 Classifiers

To build a reliable detector we need a powerful classifier. Such classifiers can be formed by a linear combination of weak classifiers, and trained with a learning algorithm to excellent classification results at a small computational cost [8,22]. In the following we explain the form of our weak classifiers.

Weak classifiers. The above described features are used to build a set of classifiers. A weak classifier is the log likelihood ratio of the probability of feature occurrence on the object with respect to the probability of feature occurrence on the non-object:

$$h_{f_a} = \ln\left(\frac{p(f_a|object)}{p(f_a|non\ object)}\right)$$

where f_a is a single feature (x, y, v_t) . Intuitively, some features occur frequently together on object but randomly together on non-object. Therefore, a better weak classifier using joint probability between two features is

$$h_{f_{ab}} = \ln\left(\frac{p(f_a, f_b|object)}{p(f_a, f_b|non\ object)}\right) \quad (2)$$

where f_a, f_b is a pair of features, which simultaneously occur on the object. The probabilities $p(f_a|object)$ and $p(f_a, f_b|object)$ and the corresponding probabilities for non-object can be estimated using multidimensional histograms of feature occurrences. Each bin in the histogram corresponds to one feature value. The probabilities are estimated by counting the feature occurrence on positive and negative examples. Some features do not appear at a particular object location which indicates a zero probability. To avoid a very large or infinite value of a weak classifier we smooth the predictions as suggested in [15].

Strong classifiers. A strong classifier is a linear combination of M weak classifiers

$$H_M(x_i) = \sum_{m=0}^M h_{f_m}(x_i)$$

where x_i is an example and the class label is $sign[H(x_i)]$. The weak classifiers h_{f_a} and $h_{f_{ab}}$ are combined using the real version of AdaBoost as proposed in [8, 15]. The error function used to evaluate the classifiers is

$$E(H_M) = \sum_i \exp[-y_i H_M(x_i)] \quad (3)$$

where y_i is a class label $[-1, 1]$ for a given training example x_i .

A strong classifier is trained separately for each of the four feature types v_t . This is motivated by the efficiency of the cascade approach. One feature type at one scale only has to be computed at a time for each cascade level. We compute features at different scales, therefore the number of strong classifiers is the number of feature types times the number of scales. The initial number of strong classifiers is therefore 20 (4 feature types at 5 scales). The number of weak classifiers used by AdaBoost depends on the scale of features and can vary from 16 to 5000.

Cascade of classifiers. The strong classifiers are used to build a cascade of classifiers for detection. The cascade starts with the best of the fastest strong classifiers. In this case the fastest classifiers are computed on the lowest scale level and

the best one corresponds to that with the lowest classification error (equation 3). Next, we evaluate all the pairs and the following classifier in the cascade is the one which leads to the best classification results. If the improvement is insignificant we discard the classifier. The number of classifiers in a cascade is therefore different for each body part. The coarse-to-fine cascade strategy leads to a fast detection. The features are computed and evaluated for an input window only if the output of the previous classifier in the cascade is larger than a threshold. The thresholds are automatically chosen during training as the minimum classifier responses on the positive training data. The output of each detector is a log likelihood map given by the sum of all strong classifiers

$$D(x_i) = \sum_{c=1}^C H_c(x_i)$$

where C is the number of strong classifiers selected for the cascade. The location of the detected object is given by a local maximum in the log likelihood map. The actual value of the local maximum is used as a confidence measure for the detection. Note that the windows classified as an object have to be evaluated by all the classifiers in the cascade. The algorithm selected 8 strong classifiers out of 20 initial for each of the face detectors and 8 classifiers for each of the head detectors (4 feature types at 2 scales). The upper body and legs detectors use 4 classifiers selected out of 20 (two feature types of gradient orientations at 2 scales).

4 Detection System

In this section we describe the detection system, that is how we find the individual parts and how we assemble them. Detection proceeds in three stages: first, individual features are detected across the image at multiple scales; second, individual parts are detected based on these features; third, bodies are detected based on assemblies of these parts.

Individual part detector. To deal with humans at different scales, the detection starts by building a scale-space pyramid by sampling the input image with the scale factor of 1.2. We then estimate the dominant orientations and compute the groups of orientations as described in section 3.1. For the profile detection we compute a mirror feature representation. The estimated horizontal and vertical orientations remain the same, only the diagonal orientations for gradient features have to be inverted. Thus, for a relatively low computational cost we are able to use the same classifiers for left and right profile views. A window of a fixed size (20×20) is evaluated at each location and each scale level of the feature image. We incorporate the feature location within the window into the feature value. This is computed only once for all the part detectors, since we use the same grid of locations for all body parts. The feature value is used as an index in a look-up table of weights estimated by AdaBoost. Each look-up table of a

body part corresponds to one strong classifier. The number of look-up tables is therefore different for each body part detector. The output of the detector is a number of log likelihood maps corresponding to the number of body parts and scales. The local maxima of the log likelihoods indicate the candidates for a body part. To detect different parts individually we threshold the confidence measure. A threshold is associated with each part and is used to make the final classification decision. However, better results are obtained by combining the responses of different part detectors and then thresholding the joint likelihood.

Joint body part detector. Given the locations and magnitudes of local maxima provided by individual detectors we use the likelihood model described in section 2.1 to combine the detection results. We start with a candidate detected with the highest confidence and larger than a threshold. This candidate is classified as a body part. We search and evaluate the candidates in the neighbourhood given by the Gaussian model of geometric relations between two parts.

For example, suppose that a head (H) is detected. This means that the log likelihood ratio

$$D_H = \log \frac{p(\mathcal{F}|H)}{p(\mathcal{F}|\text{non } H)}$$

is above threshold. We can then use the position (x_H, y_H) and scale σ_H of the detected head to determine a confidence measure that there is an upper body (U) at (x, y) with scale σ . In detail $G(x_H - x, y_H - y, \sigma_H/\sigma)$ is used to weight the computed D_U (where D_U is defined in a similar manner to D_H above). The final score is

$$D_{U|H}(x, y, \sigma) = D_U(x, y, \sigma) + G(x_H - x, y_H - y, \sigma_H/\sigma)D_H(x_H, y_H, \sigma_H) \quad (4)$$

and the upper body is detected if this score is above threshold.

A confidence measure can also be computed for more than two parts; e.g. for an upper body, head and legs (L) sub-assembly $D_{L|U,H} = D_L + G(R_{L|U})D_{U|H}$. If this score is higher than a threshold we accept this candidate as the body part and remove the closely overlapping neighbours. We can set the decision threshold higher than for the individual detectors since the confidence for body part candidates is increased with the high confidence of the other body parts. Given the new body part location we continue searching for the next one. There are usually few candidates to evaluate in the neighbourhood given by the Gaussian model.

The current implementation does not start to build the model from legs since this detector has obtained the largest classification error (cf. equation 3) and the legs are not allowed to be present alone for a body. In most of the body examples the highest log likelihood is obtained either by a face or by a head.

5 Experiments

5.1 Training Data

Each body part detector was trained separately on a different training set. Approximately 800 faces were used to train the frontal face detector and 500 faces for the profile detector. The frontal views were aligned by eyes and mouth and the profiles by eyebrow and chin. For each face example we add 2 in-plane-rotation faces at -10 and 10 degrees. To train the frontal upper body/leg model we used 250/300 images of the MIT pedestrian data base [14]. 200 images for training the profile upper body model were collected from the Internet. The initial classifiers were trained on 100K negatives example obtained from 500 images. We then selected for each body part 4000 non-object examples detected with initial classifiers. The selected examples were then used to retrain the classifiers with AdaBoost.

5.2 Detection Results

Face. The MIT-CMU test set is used to test the performance of our face detectors. There are 125 images with 481 frontal views and 208 images with 347 profiles. The combined head-face models for frontal and profile faces were used in this test. Figure 5(a) shows the face detection results. The best results were obtained with the frontal face detector using a combination of simple features and feature pairs. We obtain a detection rate of 89% for only 65 false positives. These results are comparable with state of the art detectors (see figure 5(c)). They can be considered excellent given that the same approach/features are used for all human parts. Compared to the classifiers using only single features the gain is approximately 10%. A similar difference can be observed for the profile detectors. The performance of the profile detector is not as good as the frontal one. The distinctive features for profiles are located on the object boundaries, therefore the background has a large influence on the profile appearance. Moreover the test data contains many faces with half profile views and with in-plane-rotation of more then 30 degrees. Our detector uses a single model for profiles and currently we do not explicitly deal with in plane rotations. The detection rate of 75% with only 65 false positives is still good and is the only quantitative result reported on profile detection, apart from [19].

Human. To test the upper body and legs detector we use 400 images of the MIT pedestrian database which were not used in training. 200 images containing no pedestrians were used to estimate the false positive rate. There are 10800K windows evaluated for the negative examples. The false positive rate is defined as the number of false detections per inspected window. There are $10800K/200 = 54000$ inspected windows per image. Figure 5 (b) shows the detection results for the head/face, the frontal view of the upper body part and legs as well as the joint upper body/legs model. The results for head/face are converted from figure 5(a) and displayed on 5(b) for comparison. The best results are obtained for frontal

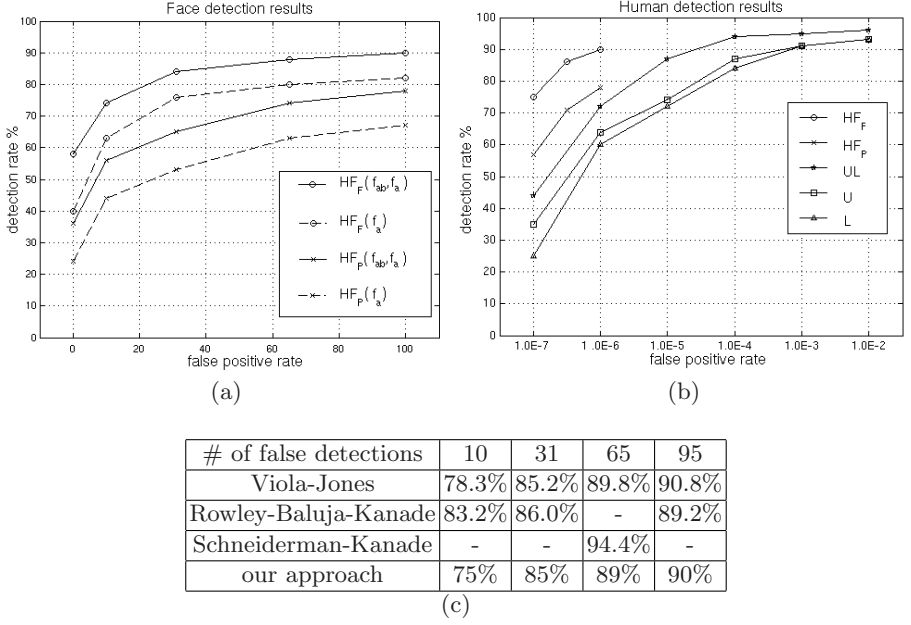


Fig. 5. (a) ROC curves for face detectors. $HF_F(f_a)$ are the results for combined frontal head H and face F detector using single features f_a . $HF_F(f_{ab}, f_a)$ are the results for the detector using both single features and feature pairs. Similarly for the profile detector HF_P . (b) ROC curves for head/face, upper body and legs detectors. The results for head/face are converted from $HF(f_{ab}, f_a)$ displayed in figure (a). U are the results for individual upper body detector and L for the individual legs detector. $U|L$ are the results for upper body detector combined with legs detector. (c) Face detection results compared to state of the art approaches.

head/face with the joint model. The result for the upper body and the legs are similar. For a low false positive rate the joint upper-body/legs detector is about 15% better than the individual upper-body and legs detectors. We obtain a detection rate of 87% with the false positive rate of 1:100000, which corresponds to one false positive per 1.8 images. This performance is better than the ones reported for pedestrian detection in [13,14]. Note that an exact comparison is not possible, since only the number of images selected for the training/test is given. In addition, our approach performs well for general configurations in the presence of occlusion and partial visibility, see figures 6 and 7.

Figure 6 illustrates the gain obtained by the joint likelihood model. The top row shows the results of the individual detectors and the bottom row the combined results. The improvement can be observed clearly. The false positives disappear and the uncertain detections are correctly classified. Some other examples are shown in Figure 7.

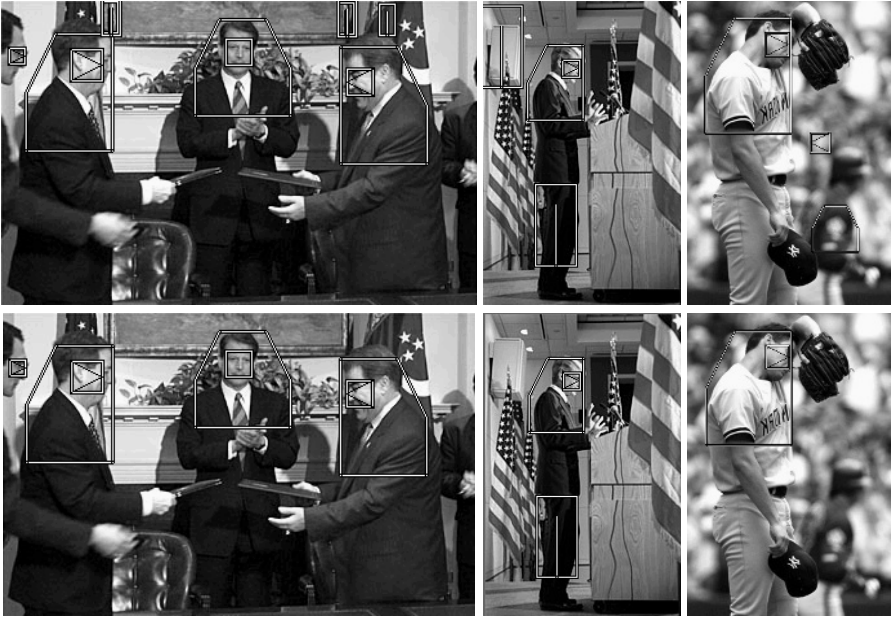


Fig. 6. Results for human detection. Top row: individual body part detection. Bottom row: detection with the joint likelihood model. The joint likelihood model significantly improves the detection results.

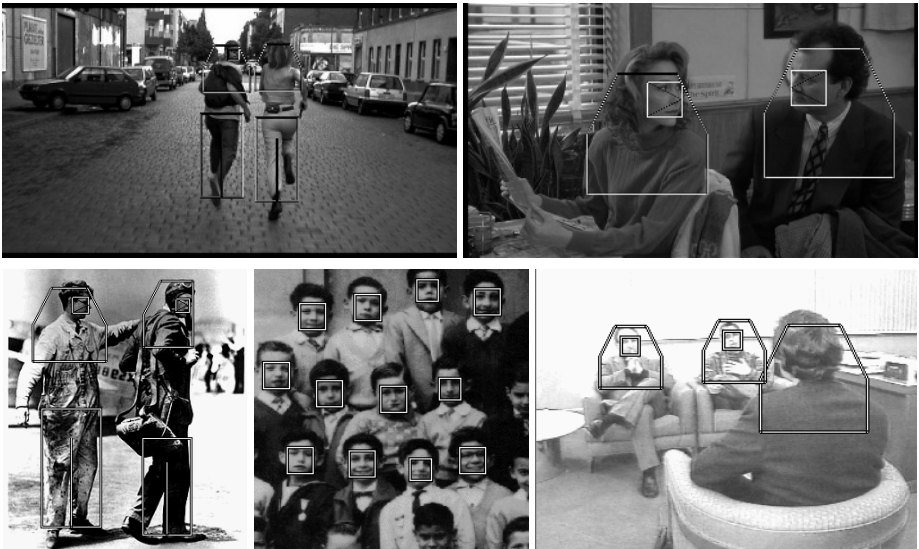


Fig. 7. Human detection with the joint model. Top row: images from movies “Run Lola Run” and “Groundhog Day”. Bottom row: images from MIT-CMU database.

6 Conclusions

In this paper we have presented a human detector based on a probabilistic assembly of robust part detectors. The key point of our approach is the robust part detector which takes into account recent advances in feature extraction and classification, and uses local context. Our features are distinctive due to encoded orientations of first and second derivatives and are robust to small translations in location and scale. They efficiently capture the shape and can therefore be used to represent any object. The joint probabilities of feature co-occurrence are used to improve the feature representation. AdaBoost learning automatically selects the best single and pairs of features. The joint likelihood of body parts further improves the results. Furthermore, our approach is efficient, as we use the same features for all parts and a coarse-to-fine cascade of classifiers. The multi-scale evaluation of a 640×480 image takes less than 10 seconds on a 2GHz P4 machine.

A possible extension is to include more part detectors, as for example an arm model. We also plan to learn more than one lower body detector. If the training examples are too different, the appearance cannot be captured by the same model. We should then automatically divide the training images in sub-sets and learn a detector for each sub-set. Furthermore, we can use motion consistency in a video to improve the detection performance in the manner of [11].

Acknowledgements. Funding for this work was provided by an INRIA post-doctoral fellowship and EC Project CogViSys.

References

1. P. Felzenszwalb. Learning models for object recognition. In *Proc. of the CVPR, Hawaii, USA*, pp. 1056-1062, 2001.
2. P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *Proc. of the CVPR, Hilton Head Island, USA*, pp. 66-75, 2000.
3. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. of the CVPR, Madison, USA*, pp. 264-271, 2003.
4. D. Forsyth and M. Fleck. Body plans. In *Proc. of the CVPR, Puerto Rico, USA*, pp. 678-683, 1997.
5. D. M. Gavrila. Pedestrian detection from a moving vehicle. In *Proc. of the ECCV, Dublin, Ireland*, pp. 37-49, 2000.
6. S. Ioffe and D. Forsyth. Probabilistic methods for finding people. *International Journal of Computer Vision*, 43(1):45-68, 2001.
7. H. Kruppa and B. Schiele. Using local context to improve face detection. In *Proc. of the BMVC, Norwich, England*, pp. 3-12, 2003.
8. S. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum. Statistical learning of multi-view face detection. In *Proc. of the ECCV, Copenhagen, Denmark*, pp. 67-81, 2002.
9. T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch - a method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283-318, 1993.

10. D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the ICCV, Kerkyra, Greece*, pp. 1150–1157, 1999.
11. K. Mikolajczyk, R. Choudhury, and C. Schmid. Face detection in a video sequence - a temporal approach. In *Proc. of the CVPR, Hawaii, USA*, pp. 96–101, 2001.
12. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proc. of the CVPR, Madison, USA*, pp. 257–263, 2003.
13. A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Transactions on PAMI*, 23(4):349–361, 2001.
14. C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.
15. Y. S. R. E. Shapire. Improving boosting algorithm using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.
16. D. Ramanan and D. A. Forsyth. Finding and tracking people from the bottom up. In *Proc. of the CVPR, Madison, USA*, pp. 467–474, 2003.
17. R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *Proc. of the ECCV, Copenhagen, Denmark*, pp. 700–714, 2002.
18. H. Schneiderman. Learning statistical structure for object detection. In *Proc. of the CAIP, Groningen, Netherlands.*, pp. 434–441, 2003.
19. H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *Proc. of the CVPR, Hilton Head Island, USA*, pp. 746–751, 2000.
20. H. Sidenbladh and M. Black. Learning image statistics for bayesian tracking. In *Proc. of the ICCV, Vancouver, Canada*, pp. 709–716, 2001.
21. L. Sigal, M. Isard, B. H. Sigelman, and M. J. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *Proc. of the NIPS, Vancouver, Canada*, 2003.
22. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of the CVPR, Hawaii, USA*, pp. 511–518, 2001.

Model Selection for Range Segmentation of Curved Objects

Alireza Bab-Hadiashar and Niloofar Gheissari

School of Engineering & Science, Swinburne University of Technology,
Melbourne, Australia
{Abab-hadiashar, ngheissari}@swin.edu.au

Abstract. In the present paper, we address the problem of recovering the true underlying model of a surface while performing the segmentation. A novel criterion for surface (model) selection is introduced and its performance for selecting the underlying model of various surfaces has been tested and compared with many other existing techniques. Using this criterion, we then present a range data segmentation algorithm capable of segmenting complex objects with planar and curved surfaces. The algorithm simultaneously identifies the type (order and geometric shape) of surface and separates all the points that are part of that surface from the rest in a range image. The paper includes the segmentation results of a large collection of range images obtained from objects with planar and curved surfaces.

1 Introduction

Model selection has received substantial attention in the last three decades due to its various applications in statistics, engineering and science. During this time, many model selection criteria (Table 1) have been proposed, almost all of which have their roots in statistical analysis of the measured data. In this paper we propose a new approach to the model selection problem based on physical constraints rather than statistical characteristics. Our approach is motivated by our observations that none of the existing model selection criteria is capable of reliably recovering the underlying model of range data of curved objects (see Figure 1).

Before we explain our model selection criterion, the problem of range segmentation for non-planar objects is briefly reviewed in the next section. Later, we propose a novel model selection tool called Surface Selection Criterion (SSC). This proposed criterion is based on the minimisation of the bending and twisting energy of a thin surface. To demonstrate the effectiveness of our proposed SSC, we devised a robust model based range segmentation algorithm for curved objects (not limited to planar surfaces). The proposed Surface Selection Criterion allows us the to choose the appropriate surface model from a library of models. An important aspect of having a correct model is obtaining the surface parameters while segmenting the surface. Recovering the underlying model is a crucial aspect of segmentation when the objects are not limited to having planar surfaces only (so more than one possible candidate exists).

Table 1. Different model selection criteria studied in this paper, N is the number of points and P is the number of parameters. J is the fisher matrix of the estimated parameters. L is equal to N . d is the dimension of manifold (here 2) and m is the dimension of the data. (here 3). f is the degrees of freedom of the assumed t distribution for MCAIC (here 1.5)

Name	Criterion
MDL[24]	$\sum_{i=1}^n r_i^2 + (P/2)\log(N)\delta^2$
GBIC[6]	$\sum_{i=1}^n r_i^2 + (Nd \log(4) + P \log(4N))\delta^2$
CP Kanatani [15]	$\sum_{i=1}^n r_i^2 + (2(dN + P) - mN)\delta^2$
CP Mallow [19]	$\sum_{i=1}^n r_i^2 + (-N + 2P)\delta^2$
GAIC[16]	$\sum_{i=1}^n r_i^2 + 2(dN + P)\delta^2$
SSD[25]	$\sum_{i=1}^n r_i^2 + (P \log(N + 2)/24 + 2 \log(p + 1))\delta^2$
CAIC[5]	$\sum_{i=1}^n r_i^2 + P(\log N + 1)\delta^2$
CAICF[5]	$\sum_{i=1}^n r_i^2 + P(\log N + 2)\delta^2 + \log J $
GMDL[16]	$\sum_{i=1}^n r_i^2 - (Nd + P)\epsilon^2 \log(\epsilon/L)^2$
MCAIC [4]	$(1 + f) \sum_{i=1}^n w_i \log \left[1 + \frac{r_i^2}{f\delta^2} \right] P(\log N + 1)\delta^2$

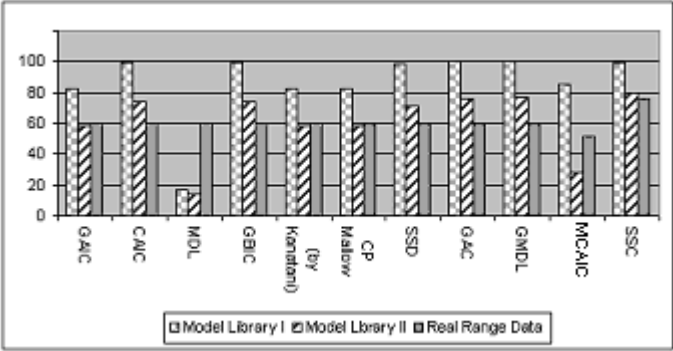


Fig. 1. Comparison of various model selection criteria for synthetic and real range data

2 Range Segmentation

A range image contains 3D information about a scene including the depth of each pixel. Segmenting a range image is the task of dividing the image into regions so that all the points of the same surface belong to the same region, there is no overlap between different regions, and the union of these regions generates the entire image.

There have been two main approaches to address the range segmentation problem. The first one is the region-based approach, in which, one groups data points so that all of them can be regarded as members of a parametric plane ([18] or [3]). The other approach is based on the edge detection and labelling edges using the jump edges (discontinuities). For example, [8] or [14].

Although the range image segmentation problem has been studied for number of years, the task of segmenting range images of curved surfaces is yet to be satisfactorily resolved. The comparative survey of Powell et al. [23] indeed reveals the challenges that need to be addressed. An early work in the area of segmentation of curved surfaces was published in 1994 by Boyer et al. [4]. They used a modified version of Bozdogan's CAIC [5] as a model selection criterion. Later, Besel and Jain [3] proposed a range image segmentation algorithm to segment range images of curved objects. The performance of their proposed algorithm for segmenting curved surfaces has been reported (by Powel et al. [23]) as unsatisfactory in many cases. Bab-Hadiashar and Suter [2] have also proposed a segmentation algorithm, which was capable of segmenting a range image of curved objects. Although they managed to segment range images into regions expressed by a specific quadratic surface, their method was limited, as it could not distinguish between different types of quadratic surfaces. Although there have been other attempts to segment range data using higher order surfaces [11,20,22,28], there have been few range segmentation algorithms capable of successfully segmenting curved objects and recovering the order of those segments. A complete literature review on range segmentation is beyond the scope of this paper. A comprehensive survey and comparison of different range segmentation algorithms was reported by Hoover et al. [13]; While a survey on model-based object recognition for range images has been reported by Arman and Aggarwal [1].

In this paper, we propose a range segmentation algorithm that is capable of identifying the order of the underlying surface while calculating the parameters of the surface. To accomplish this, we propose a new model selection criterion called Surface Selection Criterion to recover the correct surface model while segmenting the range data. An important aspect of our segmentation algorithm is that it can solve occlusion properly (it will be described in the step 6 of the segmentation algorithm). To evaluate and compare the performance of our algorithm with other existing range segmentation algorithms, we have first tested the proposed algorithm on the ABW image database [12]. The results of these experiments are shown in experimental results. Since the ABW database does not contain images of curved objects, we then created a range image database of a number of objects possessing both planar and curved surfaces. The results of those experiments (shown in Figure 1) confirm that our algorithm is not only capable of segmenting planar surfaces but can also segment higher order surfaces correctly.

3 Model Selection

In this section we propose a Surface Selection Criterion to identify the appropriate model from a family of models representing possible surfaces of a curved object. Our proposed criterion is based on minimising the sum of bending and twisting energies of all possible surfaces. Although, the bending energy of a surface has been used in the literature for motion tracking and finding parameters of deformable objects ([29] [30] [7]) and also in shape context matching and active contours([17]), it hasn't been used for model selection purposes.

3.1 Our Proposed Surface Selection Criterion

Our proposed criterion is based on minimising the sum of bending and twisting energy of all possible surfaces in a model library. To formulate our model selection criterion, we view the range data of different points of an object as hypothetical springs constraining the surface. If the surface has little stiffness, then the surface passes close to measurements (fits itself to the noise) and the sum of squared residuals between the range measurements and their associated points on the surface will be small (the sum of squared residuals in this analogy, relates to the energy of the deformed springs). However, to attain such proximity, the surface has to bend and twist in order to be close to the measured data. This in turn increases the amount of strain energy accumulated by the surface. For model selection, we propose to view the sum of bending and twisting energies of the surface as a measure of surface roughness and the sum of squared residuals as a measure of fidelity to the true data. A good model selection criterion should therefore represent an acceptable compromise between these two factors. As one may expect, increasing the number of parameters of a surface leads to a larger bending and twisting energies as the surface has more degrees of freedom and consequently the surface can be fitted to the data by bending and twisting itself so that a closer fit to measured data results (this can be inferred from the bending energy formula (Equation 1). However, the higher the number of parameters for a surface model assumed, the less the sum of squared residuals is going to be. For instance, in the extreme case, if the number of parameters is equal to the number of data points (which are used in the fitting process), then the sum of squared residuals will be zero whereas its sum of energies will be maximised.

We have a conjecture as to why this approach should be advantageous. Common statistical methods that rely essentially on probability distribution of residuals ignore spatial distribution of the deviations of data points from the surface. Whereas the above method intrinsically (through a physical model) couples the local spatial distribution of residuals to the strain energy in that locality. We argue that this is an important point as the range measurements are affected by localised factors (such as surrounding texture, surface specularities, etc) as well as by the overall accuracy and repeatability of the rangefinders.

As shown in [27], if a plate is bent by a uniformly distributed bending moment so that the xy and yz planes are the principal planes of the deflected surface, then the strain energy (for bending and twisting) of the plate can be expressed as:

$$E_{Bending+Twist} = \iint \frac{1}{2} D \left\{ \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \right)^2 - 2(1-\nu) \left[\frac{\partial^2 w}{\partial x^2} \frac{\partial^2 w}{\partial y^2} - \left(\frac{\partial^2 w}{\partial x \partial y} \right)^2 \right] \right\} dx dy$$

Equation 1

where D is the flexural rigidity of the surface and ν is Poisson's ratio (ν should be very small because in real world-objects the twisting energy in comparison with the bending energy is small). In our experiments we assume $\nu = 0.01$. We found in our experiments that the performance of SSC is not overly sensitive to the small variation of this value. In order to scale the strain energy, we divide its value by the strain energy of the model with the highest number of parameters (E_{\max}). Therefore, D will be eliminated from our computation.

To capture the trade-off between the sum of squared residuals $\sum_{i=1}^N r_i^2$ and the strain energy $E_{Bending+Twist}$, we define a function SSC such that:

$$SSC = \sum_{i=1}^N r_i^2 / N \delta^2 + P \frac{E_{Bending+Twist}}{E_{\max}}$$

where δ is the scale of noise for the highest surface (the surface with the highest number of parameters). The reason that we use the scale of noise for the highest surface (as explained by Kanatani [15]) is that the scale of noise for the correct model and the scale of noise of the higher order models (higher than the correct model) must be close for the fitting to be meaningful. Therefore, it is the best estimation of the true scale of noise which is available at this stage. The energy term has been multiplied by the number of parameters P in order to discourage choosing a higher order (than necessary) model. Such a simple measure produces good discrimination and improves the accuracy of the model selection criterion. Having devised a reasonable compromise between fidelity to data and the complexity of the model, our model selection task is then reduced to choosing the surface that has the minimum value of SSC . To evaluate our proposed Surface Selection Criterion and compare it with other well known model selection criteria (Table 1), we first created five synthetic data sets according to the surface models in Surface Library 1 (one for each model) and randomly changed the parameters of each data set 1000 times. We also added 10% normally distributed noise. The success rates of all methods in correctly recovering the underlying model are shown in Figure 1. To consider more realistic surfaces, we then considered a more comprehensive set of surface models (shown in Surface Library 2) and repeated the above experiments. The percentages of successes in this case are also shown in Figure 1.

Finally, to examine the success rate of our Surface Selection Criterion and compare it with other selection techniques on real range images, we randomly hand picked points of 100 planar surfaces of the objects in ABW range image database [12] and also 48 curved (quadratic) surfaces of our range image database. In this case, the model library used is the Surface Library 3. The results are also shown in Figure 1.

As can be seen from Figure 1, the proposed criterion (SSC) is considerably better in choosing the right model when it is applied to a variety of real range data. We should note here that the performance of MCAIC [4] is expected to be slightly better than what we have reported here if the segmentation frame work reported in [4] is used (here, we only examined the selection capability of the criteria).

Surface Library 1

Model 1	$z=ax^2+by^2+cxy+dx+ey+f$
Model 2	$z=ax^2+bxy+cx+dy+e$
Model 3	$z=ax^2+by^2+cx+dy+e$
Model 4	$z=axy+bx+cy+d$
Model 5	$z=ax+by+c$

Surface Library 2. (It should be noted that bending energy is shift and rotation invariant. Therefore there is no need to add more models to this library that have other possible combinations of x, y and z)

Model 1	$ax^2+by^2+cz^2+dxy+eyz+fx+gy+hz=1$
Model 2	$ax^2+by^2+cxy+dyz+exy+fx+gy+hz=1$
Model 3	$ax^2+by^2+cz^2+dx+ey+fz=1$
Model 4	$axy+byz+cxy+dx+ey+fz=1$
Model 5	$ax^2+by^2+cz^2+dxy+eyz+fx+gy=1$
Model 6	$ax^2+by^2+cx+dy+ez=1$
Model 7	$ax+by+cz=1$

Surface Library 3

Model 1	$ax^2+by^2+cz^2+dxy+eyz+fx+gy+hz=1$
Model 2	$ax^2+by^2+cz^2+dx+ey+fz=1$
Model 3	$ax+by+cz=1$

However, to improve the efficiency of our proposed Surface Selection Criterion, we can carry out some post processing, provided that we have a set of nested models in the model library (like Surface Library 2).

That is if the sum of squares of non-common terms between the higher surface and the next lower surface is less than a threshold, we select the lower surface. This simple step also improves the already high success rate of our proposed SSC.

4 Segmentation Algorithm

Having found a reliable method for recovering the underlying model of a higher order surface, we then proceed to use this method to perform the range segmentation of curved objects. Since our segmentation algorithm requires an estimate of the scale of noise, we have implemented the method presented in [2].

4.1 Model Based Range Segmentation Algorithm

In this section, we briefly but precisely, explain the steps of the proposed range segmentation algorithm. The statistical justification of each step is beyond the scope

of this paper but it is suffice to mention that the algorithm has been closely modelled on the ones presented in [21,26] for calculating the least median of squares. The proposed algorithm combines the above noise estimation technique and our model selection criterion SSC and delivers an effective mean for segmenting not only planar objects (as can be seen from our experiments with ABW range data) but also curved objects containing higher order models. This algorithm has been extensively tested on several range data images with considerable success (presented in experimental results). The required steps are as follows:

1. Eliminate pixels whose associated depths are not valid due to the limitation of the range finder used for measuring the depth (mainly due to specularities, poor texture, etc). These points are usually marked by the range scanner with an out-of-range number. If there are no such points we can skip this stage.
2. Find a localised data group inside the data space in which all the pixels appear on a flat plane. Even if there is no planar surface in the image, we can always approximate a very small local area (here 15×15) as a planar surface. To implement this stage and find such a data group, we choose a number of random points, which all belong to the same square of size R (this square is only for the sake of local sampling). Using these points, create an over-determined linear equation system. If the number of inliers is more than half of the size of the square, then, mark this square as an acceptable data group. The size of the square (R) is not important, however it needs to be large enough to contain adequate sample points. We set the square size as 15×15 in our experiments. We have chosen this size because a square of size 15×15 can contain enough samples. In our experiments, 30 samples were used to perform the above step.
3. Fit the highest model in the library to all the accepted data groups and find the residual for each point. Then, repeat the above two steps and accept the data group that has the least K^{th} order residual (the choice of K depends on the application [2] and is set to 10% for our experiments). This algorithm is not sensitive to the value of K . However, if we assume K to be very large, small structures will be ignored.
4. Apply a model selection method (here SSC) to the extended region (by fitting and comparing all models in the model library to the extended region) and find the appropriate model.
5. Fit the chosen model to the whole data (not segmented parts); compute the residuals and estimate the scale of noise using the technique explained in the previous section. In the next step this scale will help us to reject the outliers. It is important to note that performing this step has the advantage that it can also remedy the occlusion problem if there is any. This means that if a surface of an object is divided - occluded - by another object, we can then rightly join the separated parts as one segment.
6. Establish a group of inliers based on the obtained scale and reject the outliers. We reject those points whose squared residual is greater than the threshold T^2 multiple of the scale of noise (see the inequality $r_{n+1}^2 > T^2 \delta^2$ in the previous section). Then,

recalculate the residuals and compute the final scale using: $\delta^2 = \sum_{i=1}^n r_i^2 / (N - P)$.

7. Apply a hole-filling (here, we use a median filter of 10 by 10 pixels) algorithm to all inliers and remove holes resulting from invalid and noisy points (points where the

range finder has not been able to correctly measure the depth mainly due to their surface texture). This step is only for the sake of the appearance of the results and has no effect on the segmented surface's parameters because the fitting has already been performed. However, some of the missed invalid, and noisy points can be grouped in this step. This step is not essential and can be skipped if desired.

8. Eliminate the segmented part from the data.

9. Repeat the steps 1 to 8 until the number of remaining data becomes less than the size of the smallest possible region in the considered application.

5 Experimental Results on Segmentation

To evaluate the performance of the proposed algorithm, we have conducted a comprehensive set of experiments using real range images of various objects. The first set of experiments is solely for comparison purposes and is performed on the existing ABW database that only includes objects with planar surfaces. It is shown that the proposed technique can accurately segment the above database and its performance is similar to the best techniques presented in the literature [13]. We have then applied our technique to a set of real range images with objects having a combination of planar and curved surfaces. By these experiments we have shown that the present technique is not only capable of segmenting these objects correctly, but also truly identifies the underlying model of each surface.

5.1 ABW Image Database

In the first set of our experiments, we applied our algorithm on the ABW[12] range image database and compared our results with the ones reported by Hoover et al. [13]. As is shown here, the proposed technique is able to segment all of the images, correctly. Less than 1% of over-segmentation has occurred which is in turn resolved by using a simple merging (post-processing) step. To show the performance of our algorithm in estimating angles and comparing it with the results obtained by Hoover et al. [13], we randomly chose 100 surfaces and calculated the absolute difference between the real angle (calculated using the IDEAS CAD package) and the computed angle using the parameters of the segmented surface. The average and the standard deviation of the error for our technique and others reported in the literature are shown in Table 2. A few of the results of segmenting the ABW range image database are shown in Fig. 2.

Table 2. Comparison of accuracy of estimated angles

Technique	Angle diff. (std dev.)
USF[10]	1.6(0.8)
WSU[11]	1.6(0.7)
UB[14]	1.3(0.8)
UE[9,28]	1.6(0.9)
Proposed algorithm	1.4(0.9)

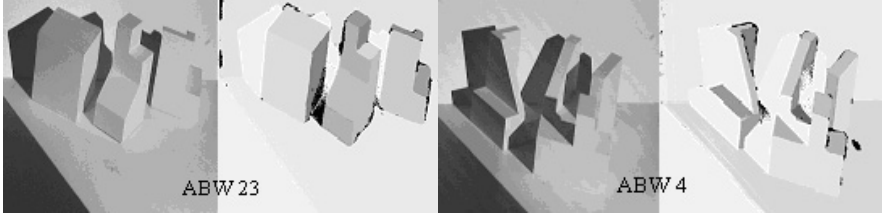


Fig. 2. Left: intensity image. Right: segmentation result

5.2 Curved Objects Database

To evaluate the performance of our algorithm in segmenting range images of curved objects, we created a range image database of a number of objects possessing both planar and curved surfaces. The actual data and their segmented results are shown in the following figures (Figure 3 to 9). We use a comprehensive model library (Surface Library 4), which consists of the most concise possible model for each object in the scene. For example, because we have cylinders (or part of) perpendicular to the xy plane in our objects, then the model $ax^2 + by^2 + cx + dy = 1$ is included in the model library (Surface Library 4).

Our segmentation algorithm was able to correctly identify the underlying model for each surface using Surface Library 4 as our model library, and SSC as a model selection criterion. The following figures shows the results of our experiments. In all of these figures, the labels show the underlying detected model. The algorithm has been successfully labeling all surfaces. For example surface 11,14,15,16, and 4 in Fig. 3 and Fig. 4, which are cylinders *perpendicular* to the xy plane, are identified to have the underlying Model 5. The underlying model for surface 25 in Fig. 6 was chosen to be Model 3, which is a cylinder *parallel* to the xy plane. Therefore our method not only can detect the cylindrical shape of the surface but it is also able to distinguish the direction of cylinders (detecting the degeneracy). For all flat surfaces SSC truly selects model 8, which represents a flat plane. An advantage of our range segmentation algorithm over region growing range segmentation algorithms is the way in which it deals with occlusion or separation of parts. Our method can detect and solve such problems correctly as can be seen from Fig. 6. In this example the planar object is located between two cylinders of the same size whose axis are co-linear. The proposed algorithm correctly detects the existence of such issues.

Surface Library 4

Model 1	$ax^2 + by^2 + cz^2 + dx + ey + fz = 1$
Model 2	$ax^2 + by^2 + cx + dy + ex = 1$
Model 3	$ax^2 + bz^2 + cx + dz + ex = 1$
Model 4	$az^2 + by^2 + cz + dx + fx = 1$
Model 5	$ax^2 + by^2 + cx + dy = 1$
Model 6	$ax^2 + bz^2 + cx + dz = 1$
Model 7	$ay^2 + bz^2 + cy + dz = 1$
Model 8	$ax + by + cz = 1$

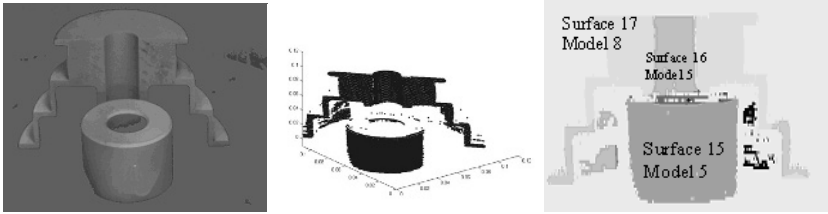


Fig. 3. Intensity image (left), plotted range data (middle) and segmentation result (right). SSC selects model 5 for the perpendicular cylinders to the xy plane (surface 16 and surface 15). The chosen model for the flat surface 17 is model 8. Surface 17, which has two separated planar parts, is an example for occlusion



Fig. 4. Intensity image (left), plotted range data (middle) and segmentation result (right). The perpendicular cylinders to the xy plane are detected correctly (model 5) using SSC. The black region illustrates the missed data. The roofs of three cylinders are in the same height, and has been correctly segmented by the proposed algorithm

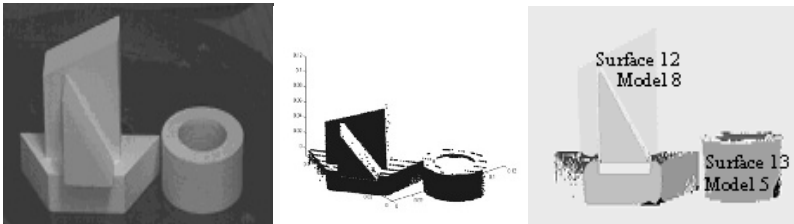


Fig. 5. Intensity image (left), plotted range data (middle) and segmentation result (right). The underlying model for surface 13, which is a cylinder perpendicular to the xy plane is selected to be model 5. For planar surfaces SSC selects model 8 as the underlying model As can be seen from the plotted rang image, despite of having noisy and invalid data, the algorithm is performed

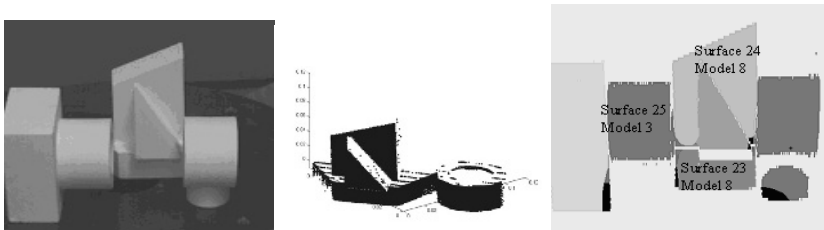


Fig. 6. Intensity image (left), plotted range data (middle) and segmentation result (right). SSC selected model 3 for the cylinders parallel to the xy plane. The underlying surface model for planar surface 24 and 23 and also other planar surfaces in the scene are chosen to be model 8

6 Conclusion

In this paper, we have proposed and evaluated a new surface model selection criterion. Using this criterion, we have also developed a robust model based range segmentation algorithm, which is capable of distinguishing between different types of surfaces while segmenting the objects. The proposed techniques both for model selection and for range segmentation has been extensively tested and have been compared with a wide range of existing techniques. The proposed criterion for model selection and the resulting segmentation algorithm clearly outperforms previously reported techniques.

References

1. Arman, F. and Aggarwal, J., Model-Based Object Recognition in Dense Range Images, *ACM Computing Surveys*, vol. 25, pp. 5–43, Mar, 1993.
2. Bab-Hadiashar, A. and Suter, D., Robust Segmentation of Visual Data Using Ranked Unbiased Scale Estimate, *ROBOTICA*, International Journal of Information, Education and Research in Robotics and Artificial Intelligence, vol. 17, pp. 649–660, 1999.
3. Besl, P. J. and Jain, R. C., Segmentation Through Variable-Order Surface Fitting., *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 167–192, Mar, 1988.
4. Boyer, K. L., Mirza, M. J., and Ganguly, G., The Robust Sequential Estimator: A General Approach and its Application to Surface Organization in Range Data, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 987–1001, 1994.
5. Bozdogan, H., Model Selection and Akaike's Information Criterion (AIC): The General Theory and Its Analytical Extensions, *Psychometrica*, vol. 52, pp. 345–370, 1987.
6. Chickering, D. and Heckerman, D., Efficient Approximation for the Marginal Likelihood of Bayesian Networks with Hidden Variables, *Machine Learning*, vol. 29, no. 2-3, pp. 181–212, 1997.
7. Duncan, J. S., Lee, F. A., Smeulders, A. W. M., and Zaret, B. L., A Bending Energy Model for Measurement of Cardiac Shape Deformity, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 307–319, 1991.
8. Fan, T. J., Medioni, G., and Nevatia, R., Segmented Descriptions of 3-D Surfaces, *IEEE Trans. on Robotics and Automation*, vol. 3, pp. 527–538, Dec, 1987.
9. Fitzgibbon, A. W., Eggert, D. W., and Fisher, R. B., High-Level CAD Model Acquisition From Range Images, *High-Level CAD Model Acquisition From Range Images*, Dept of Artificial Intelligence, Univ. of Edinburgh, 1995.
10. Goldgof, D. B., Huang, T. S., and Lee, H., A Curvature-Based Approach to Terrain Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1213–1217, 1989.
11. Hoffman, R. L. and Jain, A. K., Segmentation and Classification of Range Images, *IEEE PAMI*, vol. 9, pp. 608–620, 1987.
12. Hoover, A., <http://marathon.csee.usf.edu/range/DataBase.html>
13. Hoover, A., Jean-Baptist, G., Jiang, X., J.Flynn, P., Horst, B., Goldgof, D., Bowyer, K., Eggert, D., Fitzgibbon, A., and Fisher, R., An Experimental Comparison of Range Image Segmentation Algorithms, *IEEE Transaction on Pattern Analysis and Machine Recognition*, vol. 18, pp. 673–689, Jul, 1996.
14. Jiang, X. and Bunke, H., Range Image Segmentation: Adaptive Grouping of Edges into Region, *Proceedings of Asian Conference on Computer Vision*, Hong Kong, pp. 299–306, 1998.

15. Kanatani, K., What Is the Geometric AIC? Reply to My Reviewers, 1987(unpublished).
16. Kanatani, K., Model Selection for Geometric Inference, The 5th Asian Conference on Computer Vision, Melbourne, Australia , pp. xxi–xxxii, January 2002.
17. Kass, M., Witkin, A., and Terzopoulos, D. Snakes: Active Contour Models. In: Anonymous 1987. pp. 269–276.
18. Lee, K.-M., Meer, P., and Park, R.-H., Robust Adaptive Segmentation of Range Images, IEEE Trans. Pattern Anal. Machine Intell., vol. pp. 200–205, 1998.
19. Mallows, C. L., Some Comments on CP, Technometrics, vol. 15, pp. 661–675 , 1973.
20. Marshall, D., Lukacs, G., and Martin, R., Robust Segmentation of Primitives from Range Data in the Presence of Geometric Degeneracy, IEEE Transactions on Pattern Analysis and Machine Intelligence , vol. 23, pp. 304–314, Mar, 2001.
21. Meer, P., Mintz, D., and Rosenfeld, A., Least Median of Squares based robust analysis of Image Structure , Proce. DARPA Image Understanding Workshop, Pittsburgh, PA, pp. 231–254, Sept. 1990.
22. Newman, T. S., Flynn, P. J., and Jain, A. K., Model-based Classification of Quadric Surfaces, CVGIP: Image Understanding, vol. 58, pp. 235–249, 1993.
23. Powell, M. W., Bower, K., Jiang, X., and Bunke, H., Comparing Curved-Surface Range Image Segmentors, Proc. of 6th International Conference on Computer Vision (ICCV), Bombay, India, pp. 286–291, 1998.
24. Rissanen, J., Universal Coding, Information, Prediction and Estimation, IEEE Trans. Inf. Theory, vol. 30, pp. 629–636, 1984.
25. Rissanen, J., Modeling by Shortest Data Description, Automatica, vol. 14, pp. 465–471, 1978.
26. Rousseeuw, P. J., Least median of squares, Journal of the American Statistical Association, vol. 85, pp. 115–119, 1984.
27. Timoshenko, p. S. and Krieger, S. W. Theory of Plates and Shells. In: Chapter 2, Pure Bending of Plates, eds. Timoshenko, p. S. and Krieger, S. W. McGraw-Hill, 1959. pp. 46–47.
28. Trucco, E. and Fisher, R. B., Experiments in Curvature-based Segmentation of Range Data, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, pp. 3177–182, 1995.
29. Van Vliet, J. L. and Verbeek, W. P., Curvature and Bending Energy in Digitized 2D and 3D Images, Proceedings of 8th Scandinavian Conference on Image Analysis, Toromso, Norway , pp. 1403–1410, 1993.
30. Young, I. T., Walker, J. E., and Bowie, J. E., An analysis technique for biological shape. I. Information Control, vol. 25, pp. 357–370, 1974.

High-Contrast Color-Stripe Pattern for Rapid Structured-Light Range Imaging

Changsoo Je¹, Sang Wook Lee¹, and Rae-Hong Park²

¹ Dept. of Media Technology

² Dept. of Electronic Engineering

Sogang University

Shinsu-dong 1, Mapo-gu, Seoul 121-742, Korea

{Vision, SLee, Rhpark}@sogang.ac.kr

<http://vglab.sogang.ac.kr>

Abstract. For structured-light range imaging, color stripes can be used for increasing the number of distinguishable light patterns compared to binary BW stripes. Therefore, an appropriate use of color patterns can reduce the number of light projections and range imaging is achievable in single video frame or in “one shot”. On the other hand, the reliability and range resolution attainable from color stripes is generally lower than those from multiply projected binary BW patterns since color contrast is affected by object color reflectance and ambient light. This paper presents new methods for selecting stripe colors and designing multiple-stripe patterns for “one-shot” and “two-shot” imaging. We show that maximizing color contrast between the stripes in one-shot imaging reduces the ambiguities resulting from colored object surfaces and limitations in sensor/projector resolution. Two-shot imaging adds an extra video frame and maximizes the color contrast between the first and second video frames to diminish the ambiguities even further. Experimental results demonstrate the effectiveness of the presented one-shot and two-shot color-stripe imaging schemes.

1 Introduction

Triangulation-based structured lighting is one of the most popular ways of active range sensing and various approaches have been suggested and tested. Recently, interests have been developed in rapid range sensing of moving objects such as cloth, human face and body in one or slightly more video frames, and much attention has been paid to the use of color to increase the number of distinguishable patterns in an effort to decrease the number of structured-light projections required for ranging a scene. This paper discusses the design of stripe patterns and the selection of colors to assign to stripe illumination patterns that minimizes the effects of object surface colors, system noise, nonlinearity and limitations in camera/projector resolution for real-time range imaging in a single video frame (“one-shot”) and for near real-time imaging in double video frames (“two-shot”).

Among the systems that employ a single illumination source (projector) and a single camera, those that project sweeping laser light plane, black-and-white (BW) stripe patterns, gray-level stripes have been well investigated [1][2][3]. Since they are

projecting multiple light patterns or sweeping laser stripe, they are appropriate for stationary scenes. Hall-Holt and Rusinkiewicz have recently suggested a method based on time-varying binary BW patterns for near real-time ranging in four video frames, but object motion is assumed to be slow to keep “time coherence” [4].

Various approaches have been made to one-shot or near one-shot imaging. One class of methods is those that project continuous light patterns: Tajima and Iwakawa used rainbow pattern with continuous change of light color [5], Huang *et al.* used sinusoidal color fringe light with continuous variation of phase [6], and Carrihill and Hummel used gray-level ramp and constant illumination [7]. Although all these methods can, in principle, produce range images with high speed and high resolution only restricted by system resolution, they are highly susceptible to system noise, nonlinearity and object surface colors. Another class of approaches includes those that use discrete color patterns: Davis and Nixon designed a color-dot illumination pattern, Boyer and Kak developed color stripe patterns that can be identified by a color coding of adjacent stripes, and Zhang *et al.* also developed a color stripe pattern based on a de Bruijn sequence and stripes are identified by dynamic programming [8][9][10].

Although the color-dot and color-stripe methods are less sensitive to system noise and nonlinearity compared to those with continuous light patterns, they are also significantly affected by object color reflectance and their range resolution is limited by stripe width. Caspi *et al.* presented a three-image-frame method that can overcome the ambiguity in stripe labeling due to object surface color, but its real-time application has not been explicitly considered [11].

Most of the color stripe-based methods suggest design of color patterns that can be uniquely identified in the illumination space, but little explicit attention has been paid to the selection of colors. In this paper, we investigate the selection of colors for illumination stripe patterns for maximizing range resolution, and present a novel two-shot imaging method which is insensitive to system noise, nonlinearity and object color reflectances.

The rest of this paper is organized as follows. Section 2 describes a design of color multiple-stripe pattern, Section 3 discusses selection of colors for one-shot imaging, and Section 4 presents a method for two-shot imaging. In Section 5, generation and identification of multiple stripe patterns are discussed. Section 6 presents the experimental results and Section 7 concludes this paper.

2 Multiple-Stripe Patterns for Structured Light

A typical triangulation-based ranging system with structured light consists of an LCD or DLP projector as illustrated in Fig. 1. Design of a good light pattern is critical for establishing reliable correspondences between the projected light and camera. For real-time (30 Hz) one-shot imaging, only one illumination pattern is used and fixed in time. For two-shot imaging, on the other hand, the two light patterns should alternate in time and the projector and camera should be synchronized. Since projectors and cameras with frame rates higher than 60 Hz are now common commercially, near real-time imaging with two shots becomes much more feasible than before. In this section, we describe methods for selecting stripe patterns and colors for one-shot and two-shot imaging.

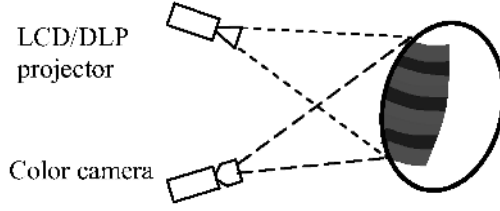


Fig. 1. Triangulation-based structured-light range imaging

The most straightforward way of generating unique color labels for M stripes would be to assign M different colors. In this case, the color distances between the stripes are small and this simple scheme can be as sensitive to system noise and nonlinearity as the rainbow pattern [5]. A small number of colors with substantial color differences are more desirable in this regard, but global uniqueness is hard to achieve due to the repeated appearance of a color among M stripes. This problem has been addressed and *spatially windowed uniqueness* has been investigated [9][10]. Instead of using one stripe for identification, we can use a small number (e.g., k) of adjacent stripes such that sub-sequence of k consecutive stripes is unique within the entire stripe sequence.

It can be easily shown that from N different colors, N^k different stripe sequences with the length k can be made. When adjacent stripes are forced to have different colors, the number of uniquely identified sequences is [9]:

$$n(N, k) = N(N-1)^{k-1} \quad (1)$$

If a single stripe is used with $N = 3$ colors, for instance, only 3 stripe labels are attainable, but the number of uniquely identifiable labels increases to 6, 12, 24 and 48 for $k = 2, 3, 4$ and 5. The entire pattern should be generated such that any sub-pattern around a stripe can be uniquely labeled.

It may be noted that with a binary BW pattern ($N = 2$) it is impossible to increase the distinct labels in this way of generating subsequences since $n(2, k) = 2 \cdot (2-1)^{k-1} = 2$. In other words, for one-shot imaging, the number of identifiable sub-patterns remains fixed regardless of the length k . Hall-Holt and Rusinkiewicz used multiple frames in time for increasing it with binary BW stripes [4].

If the color subpatterns are simply concatenated, not every stripe is uniquely identifiable [9]. We prefer having the entire stripe pattern designed such that the windows of subpatterns overlap but every stripe can be identified by a unique subpattern of k consecutive stripes centered at the stripe. Zhang *et al.* have employed this scheme using the de Bruijn sequence [10].

The design of the stripe pattern requires several choices for: the total number of stripes M , the length of the subpattern k and the number of colors N . In what follows, we turn into a discussion of criteria for choosing the stripe colors for high-resolution range imaging.

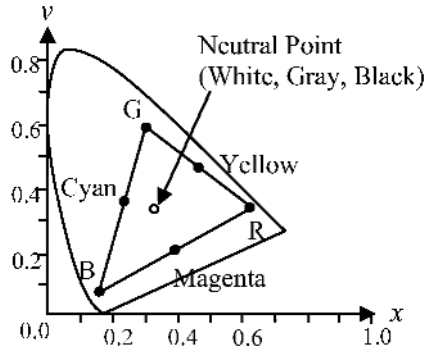


Fig. 2. CIE-xy chromaticity diagram of colors

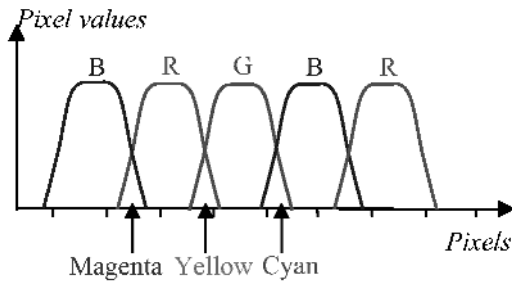


Fig. 3. Intensity profiles of RGB stripes

3 Color Selection for One-Shot Imaging

For high-resolution ranging, the stripes should be as narrow as possible. For a given resolution limit from a set of projector and camera, stripes can be better detected with higher color contrast between the stripes. The most common choice of colors has been among R, G, B, cyan (C), yellow (Y) and magenta (M), black, and white. For the stripes that appear as thin as 1.5~2 pixels in the camera, the use of some colors confound color stripe detection.

Let us first consider the case of using only deeply saturated three primary colors R, G and B, and addition of other colors later. Fig. 2 shows a chromaticity space (CIE-xy space). The colors that can be represented by the system RGB filter primaries are limited to the triangle shown in Fig. 2. For the sake of simplicity in discussion, we assume that the filter characteristics of projector and camera are identical. The image irradiance in the camera $I(\lambda)$ can be represented as:

$$I(\lambda) = g_{\theta} S(\lambda) E(\lambda), \quad (2)$$

where $S(\lambda)$ is the object reflectance, $E(\lambda)$ is the illumination from the projector, and g_{θ} is the geometric shading factor determined by the surface orientation and illumination angle. When object reflectances $S(\lambda)$ s are neutral (Fig. 4(a)), the received colors are determined by the projector illumination $E(\lambda)$, i.e., RGB stripe illumination, which

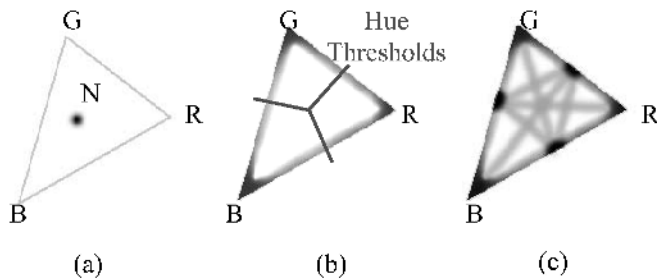


Fig. 4. Chromaticity diagram: (a) neutral object reflectances, (b) dispersed chromaticities under RGB stripe illumination and (c) dispersed chromaticities under RGBCMY stripe illumination

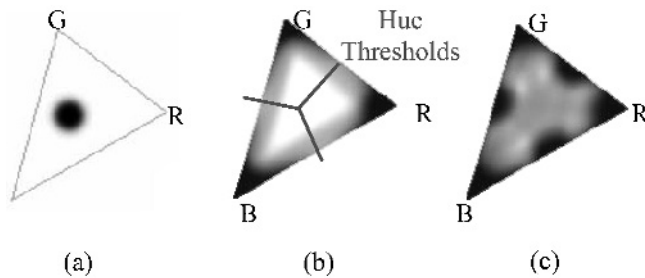


Fig. 5. Chromaticity diagram: (a) colored object reflectances, (b) dispersed chromaticities under RGB stripe illumination and (c) dispersed chromaticities under RGBCMY stripe illumination

splits the reflections from the neutral surfaces to the RGB points, i.e., the vertices in the chromaticity triangle (Fig. 4(b)). The one-dimensional hue value can be used effectively for separating the distinct colors. The more saturated the three stripe-illumination colors are, the farther apart the image stripe chromacities are and therefore the easier to detect against each other.

For contiguous RGB stripes, other colors than RGB appear in the image from the linear combination of RGB colors due to the limited bandwidth of the camera/projector system as illustrated in Fig. 3. The linear combinations of RG, GB and BR colors are generated at the boundaries. Fig. 4(a) shows neutral reflectance in the chromaticity triangle, and 4(b) shows the spread of the chromaticities at the boundaries. With only RGB illumination colors, thresholding by hue is effective in the presence of false boundary colors as illustrated in Fig. 4(b). Hue thresholding also works well for the reflections from moderately colored object surfaces as illustrated in Fig. 5. Fig. 5(a) and (b) depict the chromaticities of colored objects and spread chromaticities under RGB stripe illumination, respectively.

When the stripe width is 1.5~2 pixels, the pixels with the false colors around the stripe boundaries is substantial compared to the stripe colors. When additional colors such as CMY are used, the false boundary colors significantly confound stripe detection and identification since the CMY colors are linear combinations of the RGB primaries. The false colors can easily break subpatterns. Fig. 4(c) and 5(c) depict the chromaticity spreads under RGBCMY illumination from neutral and colored objects, respectively. It can be seen that additional false colors appear and there is no easy way to separate them from the stripe colors.

We find that for high-resolution imaging, the use of only RGB colors results in the best resolution and the smallest errors. This restricts the number of color code N to 3. White can be added if surface shading is insignificant and black might be added when ambient light is low.

Other 3-color primaries such as CMY would perform similarly to RGB only if the chromaticity distances of the primaries were as large as those of RGB. However, the color distances between CMY (as synthesized from the linear combinations of RGB) are substantially smaller. (See Fig. 2.) The CMY colors in Fig. 2 have substantially less saturation than RGB. The only way to keep the CMY distances comparable to those of RGB is to employ narrowband CMY color filters in the camera instead of RGB, but such commercial cameras are rare in reality.

If object surfaces have substantially saturated colors, the reflectance chromaticities are dispersed so widely that even strong RGB stripes cannot separate them for detection and extra information is needed.

4 Two-Shot Imaging Method

When synchronized high-speed camera and projector are available, more than one frame can be used to discount the effects of highly saturated object colors and ambient illumination in range sensing. Many commercial cameras and projectors offer external trigger and frame refresh rate higher than 60 Hz. We present a two-shot imaging method that uses two video frames and stripes with highly saturated projector colors.

When projection of two light colors $E_1(\lambda)$ and $E_2(\lambda)$ alternates in time, the following two images can be obtained:

$$\begin{cases} I_1(\lambda) = g_\theta S(\lambda)[E_1(\lambda) + A(\lambda)] \\ I_2(\lambda) = g_\theta S(\lambda)[E_2(\lambda) + A(\lambda)] \end{cases}, \quad (3)$$

where the effect of ambient illumination is included. Since it is assumed that objects are stationary during two consecutive video frames in a short time interval, g_θ and $S(\lambda)$ are common to both the images.

Caspi *et al.* used an extra image with the projector turned off to estimate the influence of the ambient illumination from $I_A(\lambda) = g_\theta S(\lambda)A(\lambda)$ [11]. After discounting $A(\lambda)$, the ratio of the images will be dependent only on the designed illumination colors without any influence of surface color and shading, i.e.,:

$$\frac{I_2(\lambda) - I_A(\lambda)}{I_1(\lambda) - I_A(\lambda)} = \frac{E_2(\lambda)}{E_1(\lambda)}. \quad (4)$$

With the commonly used assumption of spectral smoothness of $S(\lambda)$ in each color channel and some appropriate color calibration described in [11], the responses in the color channels can be decoupled and analyzed independently with the following ratios:

$$\frac{R_2}{R_1}, \quad \frac{G_2}{G_1} \quad \text{and} \quad \frac{B_2}{B_1}.$$

While this is an effective way of discounting objects colors if image values are in the linear range of projector and camera responses and many combinations of color ratios can be produced for stripes, the ratios become unstable when I_1 and I_2 are small due to small g_θ and $S(\lambda)$ values and when image values are clipped on highly reflective surfaces. The geometric factor g_θ is small on shaded surfaces whose surface orientation is away from the illumination angle. A “three-shot” method by Caspi *et al.* uses an extra shot to measure the ambient illumination $I_A(\lambda)$ for its compensation and relies on the color ratios [11]. However, the color ratios are highly unstable for bright and dark regions.

Instead of assigning many RGB projector colors and identifying them in the linear range, we seek a small number of stripe signatures from the sign of color difference to reduce the sensitivity to light intensity and ambient light without the third image for the estimation of ambient light. From Equation 3, the difference of two images is given as:

$$\begin{aligned}\Delta I(\lambda) &= I_2(\lambda) - I_1(\lambda) \\ &= g_\theta S(\lambda)[E_2(\lambda) - E_1(\lambda)], \\ &= g_\theta S(\lambda)\Delta E(\lambda)\end{aligned}\tag{5}$$

and the ambient illumination is discounted. In Equation 5, it can be seen that though $\Delta I(\lambda)$ is significantly affected by g_θ and $S(\lambda)$, its sign is not since g_θ and $S(\lambda)$ are both positive. We can derive a few stripe labels from the sign of image difference.

When color channels are decoupled with the same color assumption described above, the differences of RGB channel values are:

$$\begin{cases} \Delta R = g_\theta S^R \Delta E^R \\ \Delta G = g_\theta S^G \Delta E^G \\ \Delta B = g_\theta S^B \Delta E^B \end{cases},\tag{6}$$

where S^R , S^G and S^B are the object reflectances in R, G and B channels, respectively, and ΔE^R , ΔE^G and ΔE^B are the intensity differences of projection illumination in R, G and B channels, respectively. From the positive and negative signs of ΔR , ΔG and ΔB , we can obtain $2^3=8$ distinct codes to assign in a projector stripe. If we construct subpatterns with $N=8$, $n(N,k) = 8 \cdot 7^{(k-1)}$ unique subpatterns can be generated according to Equation 1. In the design of subpatterns, however, it was observed that the spatial transition of colors over the stripes in multiple channels make false colors due to the inconsistency of color channels and those false colors are not easily identifiable. If we allow spatial color transition only in one of the RGB channels, only three types of transitions are allowed from one stripe to another. In this case, it can be shown that the number of unique subpatterns for the length k in two-shot, “ $n2$ ” is given as:

$$n2(m,k) = 2^m m^{k-1}\tag{7}$$

where m is the number of possible spatial color transitions, e.g., $m=3$ for RGB colors and $m=2$ for GB colors. To maximize the ΔR , ΔG and ΔB values for good discriminability between the positive and negative signs, the intensity difference of projection color should be maximized; we assign the minimum and maximum programmable values for the frame 1 and 2.

The channel value differences in Equation 6 are also affected by small values of g_θ and $S(\lambda)$, but we claim that their influence is much smaller with the difference than the ratio for given system noise. Furthermore, the effect of system nonlinearity and RGB color clipping is much less pronounced with the signs of the difference. To demonstrate the efficacy of this approach, we use only presented method in our experiments with only two color channels, G and B, in our experiments, i.e., with $N=2^2=4$ codes.

5 Multiple-Stripe Pattern Synthesis and Identification

The requirements for a program that generates the subpatterns are as follows:

- (a) Different colors or codes should be assigned to adjacent stripes to make distinguishable stripes.
- (b) The subpattern generated by the i th stripe should be different from any subpatterns generated up to the $(i-1)$ th stripe.

For the two-shot imaging, the following extra requirements should be added:

- (c) Only one channel among RGB should make a color transition between the adjacent stripes.
- (d) Colors in each stripe in the second frame should be the reverse of those in the first frame in each channel. The reverse of maximum value is the minimum value and *vice versa*.

There is a tradeoff to make between the number of total stripes to cover a scene M and the number of stripes in a whole pattern $n(N, k)$. For high-resolution imaging M should be kept large and $n(N, k)$ should be close to M for a unique encoding of all the stripes. Otherwise, the whole pattern should be repeated for the scene, and its subpatterns are not globally unique. This means that for small N , the length of the subpattern k should be large. Wide subpatterns, however, are not reliable near the object boundaries and occlusions. The best compromise we make for one shot imaging (“ $n1$ ”) is to have $n1(3, 7)=192$ with $k=7$ for $M \approx 400$ and let the pattern appear twice. Stripe identification or unwrapping is not difficult since identical subpatterns appear only twice and they are far apart in the image. For the two-shot imaging with only GB channels, $n2(2, 7) = 2^2 \cdot 2^{7-1} = 256$ with $k=7$. The generated patterns for one-shot and two-shot image are shown in Figs. 6 and 7.

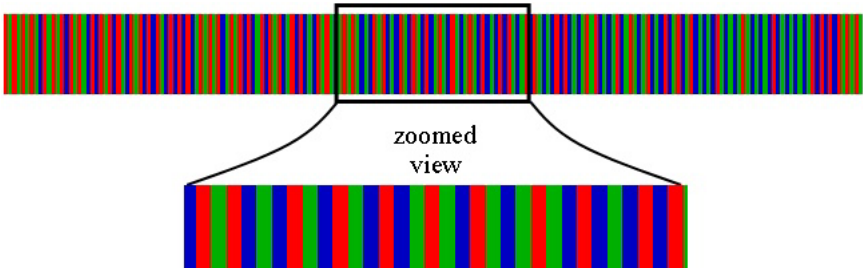


Fig. 6. RGB pattern for one-shot imaging: 192 unique subpatterns

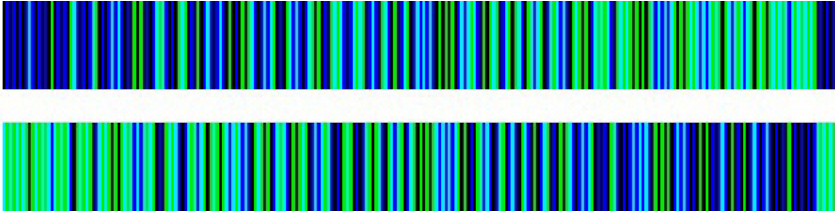


Fig. 7. GB patterns for two-shot imaging: 256 unique subpatterns in each frame

Stripe segmentation in the received image can be done mostly based on two methods: stripe color classification and edge detection by color gradients. Both should work equivalently for imaging with thick stripes. For high frequency stripe imaging, however, color gradients are not stably detected after smoothing and we prefer stripe color classification.

6 Experimental Results

We have carried out experiments with a number of objects using the described one-shot and two-shot methods. We used a Sony XC-003 3-CCD 640×480 color camera, and an Epson EMP-7700 1024×768 projector. The encoded patterns are projected onto the various objects by the projector, and the camera captures the scene. We did not use any optimization algorithm for extracting stripe identification labels since the direct color classification and decoding algorithms work well. The projected patterns all consist of stripes with one-pixel width, and the width of stripes in the captured images is around 2 pixels. We used RGB stripes with $k=7$ and $n1=192$ for one-shot imaging and GB stripes with $k=7$ and $n2=256$ for two-shot imaging, as described in Section 5. We used only 2 color channels because they keep $n2$ large enough, and the reason why we chose GB instead of RG is that the G and B color channels have slightly less crosstalk than the R and G channels in our camera-projector setup.

Fig. 8 shows the results from a human face with one-shot and two-shot imaging. For one-shot imaging, Figs. 8(a), (d), (b) and (f) show the subject under white projector illumination, stripe pattern projection, pseudo-color display of identified stripes from subpatterns, and the range image from the identified stripes, respectively. For two-shot imaging, Fig. 8(e), (c) and (g) show the two stripe patterns, pseudo-color display of identified stripes from subpatterns, and the range image from the identified stripes, respectively. Since the face has moderate colors, both the methods work well.

One-shot and two-shot imaging has also been tested with a flat color panel with highly saturated color patches. Fig. 9(a), (b), (c) and (d) shows the color panel under white light, one of the color patterns for two-shot imaging, the range image from one-shot imaging, and the range image from two-shot imaging, respectively. It can be seen that the strong colors of surface reflectance confound the one-shot imaging significantly. Fig. 10(a), (b), (c) and (d) show chromaticity plot from the human face, those from the color panel under white light, and from RGB stripe light, and the plot of panel colors in GB space from two-shot imaging.

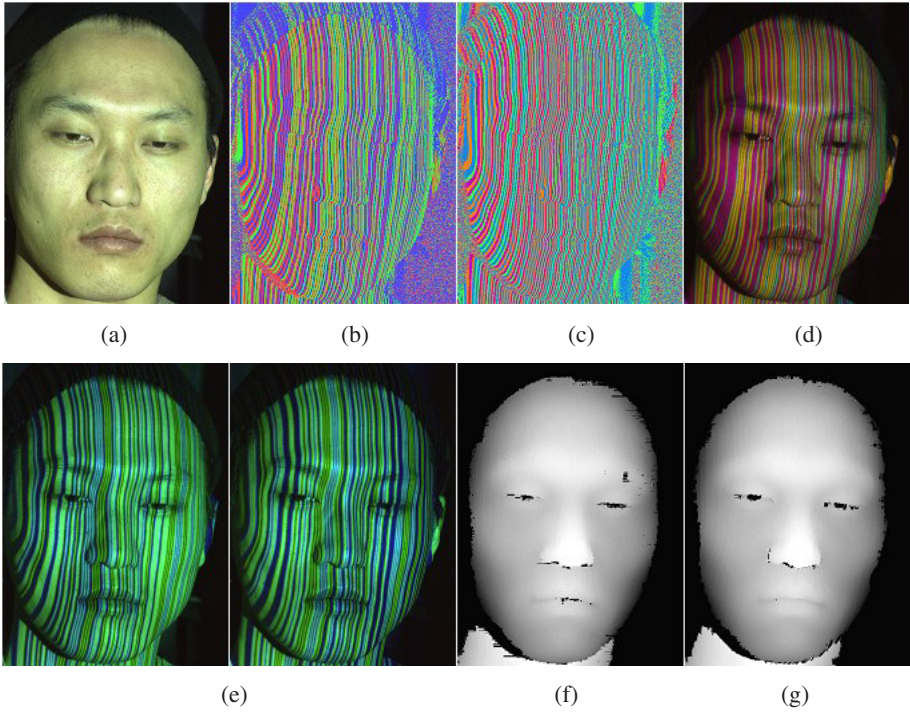


Fig. 8. Results from a human face: (a) A human face under white light, (b) identified stripes from subpattern from one-shot imaging (pseudo color assignment for each stripe), (c) from two-shot imaging (pseudo color assignment for each stripe), (d) color stripes for one-shot imaging, (e) two stripe patterns for two shot imaging, (f) range image from one-shot imaging, and (g) range image from two-shot imaging

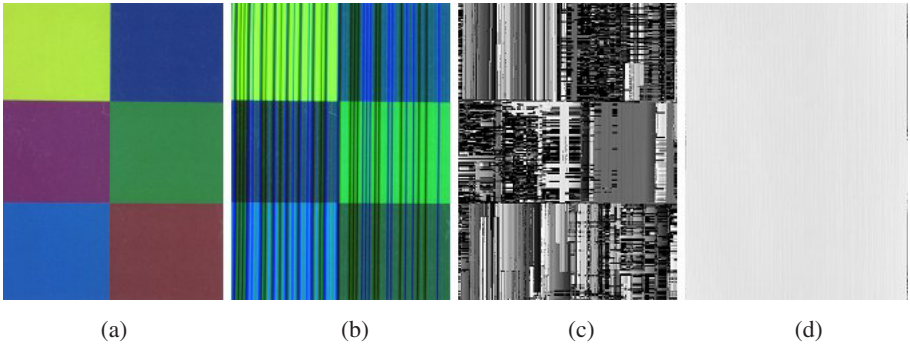


Fig. 9. Experiments with a color panel: (a) under white light, (b) one of the color patterns for two-shot imaging, (c) range image from one-shot imaging, and (d) range image from two-shot imaging

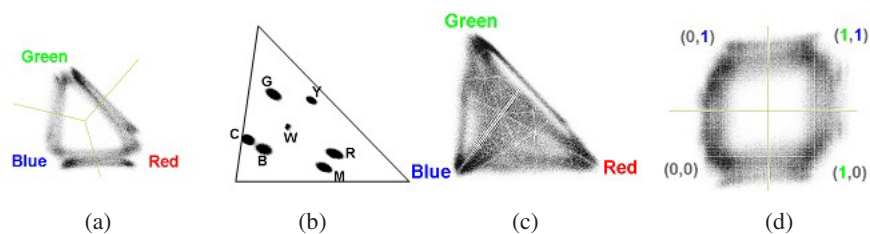


Fig. 10. Color plots: (a) Chromaticities from the human skin under RGB stripe light, (b) those from the color panel under white light, and (c) those from RGB stripe light, and (d) plot of panel colors in GB space from two-shot imaging

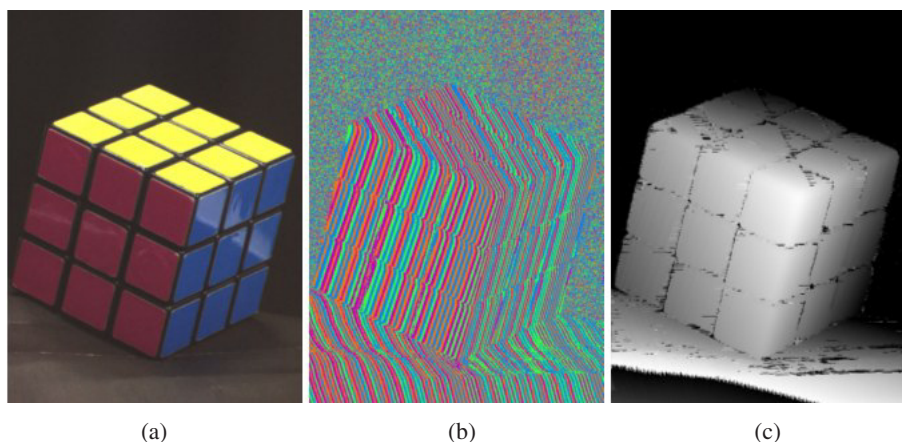


Fig. 11. Two-shot imaging with Rubik's cube (yellow, nearly-red and blue surfaces): (a) the object, (b) stripe segmentation by GB differences and (c) its range result

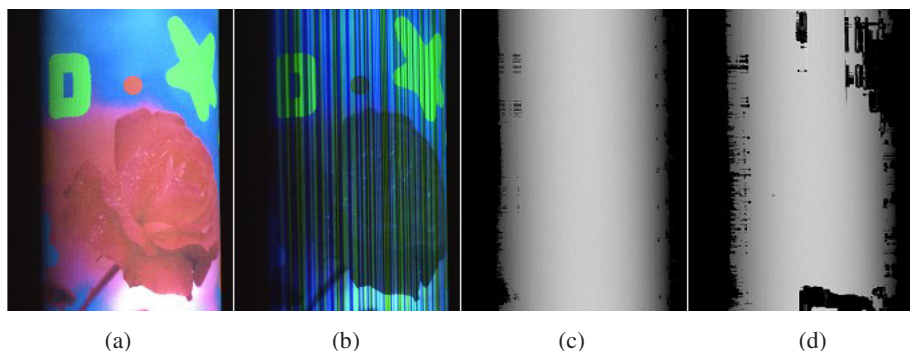


Fig. 12. Two-shot imaging with a cylindrical object: (a) under white illumination, (b) one of the two stripe patterns, (c) range image with the presented method, and (d) with the method in [11]

As can be seen from Fig. 10(a), the face colors are moderate and good for the one-shot imaging. However, the colors from the panel are so strong that projection of RGB colors cannot reliably separate the colors of stripe-like regions on the captured image into the different regions in the chromaticity as shown in (b) and (c), while we can easily identify GB-combination codes in (d).

Fig. 11 shows (a) a strongly colored object (Rubik's cube), (b) its stripe segmentation and (c) the high-resolution range result using the presented two-shot method. The stripes are properly segmented in spite of the strong surface colors and specular reflections, and even so is the black paper region at the bottom. Note that the range result is sufficiently good despite the discontinuities of the multiple-stripe sequence in the black lattice regions. Fig. 12 compares the result from a colored cylinder with the proposed two-shot method and that with the method of Caspi *et al.* in [11]. It can be seen that the presented method has the advantage in the bright (or highly saturated) and dark regions.

7 Conclusion

For rapid range sensing, we described a design of multiple-stripe patterns for increasing the number of distinguishable stripe codes, discussed a color selection scheme for reducing the ambiguity in stripe labeling in one-shot imaging, and presented a novel method for two-shot imaging that is insensitive to object color reflectance, ambient light and limitations in projector/sensor resolution. We showed that maximizing color contrast between the stripes in one-shot imaging reduce the ambiguities resulting from system resolution and object colors to some degree, and the new method of utilizing color differences in two shot imaging further reduce the ambiguities resulting from colored object surfaces, ambient light and sensor/projector noise and nonlinearity. By using the signs of color differences instead of color ratios in two-shot imaging, we can obtain more reliable information in the bright, dark, and strongly-colored regions of objects and also minimize the number of shots in multiple-frame imaging.

Acknowledgement. This work was supported by grant number R01-2002-000-00472-0 from the Basic Research program of Korea Science & Engineering Foundation (KOSEF).

References

1. M. Rioux, G. Bechthold, D. Taylor, and M. Duggan, "Design of a large depth of view three-dimensional camera for robot vision," *Optical Engineering*, 26(12):1245–1250, Dec 1987
2. K. Sato and S. Inokuchi, "Three-dimensional surface measurement by space encoding range imaging," *Journal of Robotic System*, 2:27–39, 1985
3. E. Horn and N. Kiryati, "Toward optimal structured light patterns," *Image and Vision Computing*, 17, 1999

4. O. Hall-Holt and S. Rusinkiewicz, "Stripe Boundary Codes for Real-time Structured-light Range Scanning of Moving Objects," Proc. ICCV, 2001
5. J. Tajima and M. Iwakawa, "3-D Data Acquisition by Rainbow Range Finder," Proc. 10th ICPR, pp. 309-313, Atlantic City, N.J., 1990
6. P.S. Huang, Q. Hu, F. Jin and F. Chiang, "Color-encoded digital fringe projection technique for high-speed three-dimensional surface contouring, SPIE Optical Engineering, vol. 38(06), pp.1065–1071, 1999
7. B. Carrihill and R. Hummel, "Experiments with the intensity ratio depth sensor," Computer Vision, Graphics, and ImageProcessing, 32:337–358, 1985
8. C. J. Davies and M. S. Nixon, "A hough transformation for detecting the location and orientation of three-dimensional surfaces via color encoded spots," IEEE Trans. on Systems, Man, and Cybernetics, 28(1B), 1998
9. K. L. Boyer and A. C. Kak, "Color-encoded structured light for rapid active ranging," IEEE Transactions on Pattern Analysis and Machine Intelligence 9 (1987), no. 1, 14–28
10. L. Zhang, B. Curless, and S. M. Seitz, "Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming," 1st international symposium on 3D data processing, visualization, and transmission, Padova, Italy, June 19-21, 2002
11. D. Caspi, N. Kiryati and J. Shamir, "Range Imaging with Adaptive Color Structured Light," IEEE PAMI, Vol. 20, pp. 470–480, 1998

Using Inter-feature-Line Consistencies for Sequence-Based Object Recognition

Jiun-Hung Chen^{1*} and Chu-Song Chen²

¹ University of Washington, Seattle, WA 98195, USA,
jhchen@cs.washington.edu,

² Institute of Information Science, Academia Sinica, Taipei, Taiwan
song@iis.sinica.edu.tw

Abstract. An image sequence-based framework for appearance-based object recognition is proposed in this paper. Compared with the methods of using a single view for object recognition, inter-frame consistencies can be exploited in a sequence-based method, so that a better recognition performance can be achieved. We use the nearest feature line method (NFL) [8] to model each object. The NFL method is extended in this paper by further integrating motion-continuity information between features lines in a probabilistic framework. The associated recognition task is formulated as maximizing an *a posteriori* probability measure. The recognition problem is then further transformed to a shortest-path searching problem, and a dynamic-programming technique is used to solve it.

1 Introduction

Appearance-based methods [2][9][10][12][13][14][15][19] emphasize the use of view-based representations of objects, which are constructed from a set of views of an object in a pre-processing (or learning) stage, for object recognition or tracking. The collection of views is usually recorded in a compact way through principle component analysis (PCA) [10][12], support vector machine (SVM) [13][14] or neural networks [15][19]. In the past, Murase and Nayar [10] observed that all the training feature vectors (e.g., vectors in association with a PCA representation) of an object consist of a manifold in the feature space. They approximated the manifold by using a spline interpolation for the feature vectors of a set of sampled views. In addition, Roobaert and van Hulle [14] used SVM and Roth, Yang and Ahuja [15] used sparse network of winnows (SNoW) for modelling the sampled views. Appearance-based techniques can also be used for recognizing objects in a cluttered environment [12][13] and for tracking long image sequences or sequences across views [2].

Object recognition via linear combination [16][8][1][17] is an interesting and informative concept received considerable attentions in recent years. In [16], Ullman and Basri demonstrated that the variety of views depicting the same object

* This work was done while he was a research assistant at Institute of Information Science, Academia, Taipei, Taiwan.

under different transformations can often be expressed as the linear combination of a small number of views, and suggested how this linear combination property may be used in the recognition process. In [17], Vetter and Poggio proposed a method based on linear object classes for synthesizing new images of an object from a single 2D view of the object by using corresponding feature points. In addition to recognizing or synthesizing views of different poses, linear combination has also been shown very useful for visual recognition or view synthesis under different illumination conditions. For example, Belhumeur and Kriegman [1] have shown that a new image under all possible illumination conditions can be expressed as a linear combination of some basis images formed by a convex polyhedral cone in R^n if the illumination model is Lambertian.

Recently, a linear method called the nearest feature line (NFL) was proposed for object recognition [8][7]. It uses the collection of lines passing through each pair of the feature vectors belonging to an object to model appearances of this object. Instead of using splines [10], the NFL method uses a linear structure to represent the appearance manifold, which has a close relationship with the linear-combination approaches mentioned above. In essence, infinite feature vectors of the object class can be generated from finite sample vectors with the NFL method. Note that NFL can be treated as an extension of the nearest-neighbor (NN) method, and it has been theoretically proven that the NFL method can achieve a lower probabilistic error than NN when the number of available feature points for each object class is finite and the dimension of a feature space is high [20]. An experimental evaluation of the NFL method in image classification and retrieval was given in [7], which shows that it can make efficient use of knowledge about multiple prototypes of a class to represent that class.

In this paper, a framework for sequence-based object recognition is proposed by employing the concept of feature lines. In particular, by further considering inter-feature-line consistencies, our method can use several images of an object as the training input for building an image-sequence-based recognition system. More specifically, the database built in our work contains information about possible moves between views in the database. Therefore, it contains much more information than an unordered set of views, and our method would only be applicable to problem domains where this information is available. The main idea of our method is that it tries and finds objects that not only match the individual images, but also makes sure that the sequence of views in the query could match a similar sequence of views in the database. In other words, the input database does not consist of isolated example images; but rather that these images are related to each other via motion consistency. Hence, our method would work much better as more images are added, which means that the performance improvement over adding more images would be relatively better for this method than for other methods. In our framework, a recognition task is formulated as a problem of maximizing an *a posteriori* probability measure. This problem is further reduced to a most-probable-path searching problem in a specially designed graph, which can be effectively solved with dynamic programming.

This paper presents a general framework for sequence-based object recognition, which can be used for real-world applications such as face recognition.

For example, by incorporating our recognition method with existing face detection and tracking algorithms [4][18], this framework can be used to achieve sequence-based face recognition for person identity verification. The remainder of this paper is organized as follows. In Section 2, we present a probabilistic formulation for sequence-based object recognition by employing inter-feature-line consistencies. Section 3 gives the main algorithm of this paper. Some experimental results are shown and discussed in Section 4. Finally, we make conclusions in Section 5.

2 Object Recognition by Using Inter-feature-Line Consistency

In the following, we will first review the nearest-feature-line (NFL) method [7][8] in Section 2.A. Then, we will characterize inter-feature-line consistencies formulated in our work in Section 2.B.

2.1 Approximate Manifold with Feature Lines – An Introduction

Assume that we have M objects, and let $X^c = \{\mathbf{x}_i^c | i = 1, \dots, N_c\}$ be a set of N_c training feature vectors belonging to object $c, c = 1, 2, \dots, M$. A *feature line* (FL) $\overline{\mathbf{x}_i^c \mathbf{x}_j^c}$ ($i \neq j$) of object c is defined as a straight line passing through \mathbf{x}_i^c and \mathbf{x}_j^c . A FL space of object c is denoted by $S^c = \{\overline{\mathbf{x}_i^c \mathbf{x}_j^c} | 1 \leq i, j \leq N_c, i \neq j\}$, where $\overline{\mathbf{x}_i^c \mathbf{x}_j^c} = \overline{\mathbf{x}_j^c \mathbf{x}_i^c}, 1 \leq i, j \leq N_c$, and the number of feature lines in S^c , denoted by K_c , is $\frac{N_c(N_c-1)}{2}$. When there are M classes in the database, M such FL spaces can therefore be constructed, composed of a total number of $N_{total} = \sum_{c=1}^M K_c$ FLs. Let $\Gamma = S^1 \cup S^2 \cup \dots \cup S^M$ be the collection of all N_{total} feature lines. The *FL distance* from a query \mathbf{q} to some feature line $\overline{\mathbf{x}_i \mathbf{x}_j}$ ($i \neq j$) is defined as

$$d(\mathbf{q}, \overline{\mathbf{x}_i \mathbf{x}_j}) = \|\mathbf{q} - \mathbf{p}\| \quad (1)$$

where $\|\cdot\|$ is the 2-norm and \mathbf{p} is the projection point of the query \mathbf{q} onto $\overline{\mathbf{x}_i \mathbf{x}_j}$. The projection point \mathbf{p} can be computed as $\mathbf{p} = \mathbf{x}_i + \mu(\mathbf{x}_j - \mathbf{x}_i)$ and the position parameter $\mu \in R$ is

$$\mu = \frac{(\mathbf{q} - \mathbf{x}_i)^t(\mathbf{q} - \mathbf{x}_i)}{(\mathbf{x}_j - \mathbf{x}_i)^t(\mathbf{x}_j - \mathbf{x}_i)}. \quad (2)$$

Figure 1 (a) illustrates FLs and FL distances. Note that when $0 \leq \mu \leq 1$, \mathbf{p} is an interpolating point between \mathbf{x}_i and \mathbf{x}_j . Otherwise, \mathbf{p} is an extrapolating point either on the \mathbf{x}_i side (when $\mu > 1$) or the \mathbf{x}_j side (when $\mu < 0$). NFL recognizes \mathbf{q} as object c^* by computing the minimal FL distance between all features lines contained in Γ as shown below.

$$\begin{aligned} d(\mathbf{q}, \overline{\mathbf{x}_{i^*}^{c^*} \mathbf{x}_{j^*}^{c^*}}) &= \min_{1 \leq c \leq M} \min_{\overline{\mathbf{x}_i^c \mathbf{x}_j^c} \in S^c} d(\mathbf{q}, \overline{\mathbf{x}_i^c \mathbf{x}_j^c}) \\ &= \min_{\overline{\mathbf{x}_i^c \mathbf{x}_j^c} \in \Gamma} d(\mathbf{q}, \overline{\mathbf{x}_i^c \mathbf{x}_j^c}) \end{aligned} \quad (3)$$

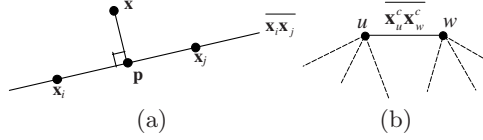


Fig. 1. (a) The concepts of a FL $\overline{x_i x_j}$ and the FL distance from a query x to $\overline{x_i x_j}$. (b) The concept of neighboring FLs. The feature lines drawn with dash lines are neighboring FLs of $\overline{x_u^c x_w^c}$.

2.2 Inter Feature-Line Consistencies

In essence, the NFL method [8][7] only uses the distance-from-manifold information for object recognition. In our framework, we further extend the NFL method to a new one where inter-feature-line consistencies are incorporated by using the concept of neighboring FLs. Given a feature line $\overline{x_u^c x_w^c}$, we denote $\Psi_{c,u,w} = \{\overline{x_u^c x_w^c}\} \cup \{\overline{x_u^c x_{w_1}^c} \mid 1 \leq w_1 \leq N_c, w_1 \neq u, w_1 \neq w\} \cup \{\overline{x_{u_1}^c x_w^c} \mid 1 \leq u_1 \leq N_c, u_1 \neq w, u_1 \neq u\}$ to be the set of its *neighboring FLs*. An illustration of neighboring FLs is shown in Figure 1(b). The number of neighboring FLs of $\overline{x_u^c x_w^c}$ is therefore equal to $2N_c - 3$.

Let $\Lambda_q = \{\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_L\}$ be a sequence of $L + 1$ views. In addition, let $\Psi = \{\mathbf{l}_i \in \Gamma \mid i = 0, 1, \dots, L\}$ be a set of $L + 1$ FLs where each \mathbf{q}_i matches its projection point \mathbf{p}_i on \mathbf{l}_i . We recognize Λ_q by finding Ψ^* that maximizes the following *a posteriori* probability:

$$\begin{aligned}
 \Psi^* &= \arg \max_{\Psi} P(\Psi | \Lambda_q) \\
 &= \arg \max_{\Psi} P(\mathbf{l}_0, \dots, \mathbf{l}_L | \mathbf{q}_0, \dots, \mathbf{q}_L) \\
 &= \arg \max_{\Psi} P(\mathbf{q}_0, \dots, \mathbf{q}_L | \mathbf{l}_0, \dots, \mathbf{l}_L) P(\mathbf{l}_0, \dots, \mathbf{l}_L) \\
 &= \arg \max_{\Psi} P(\mathbf{l}_0) P(\mathbf{q}_0 | \mathbf{l}_0) \prod_{i=0, \dots, L-1} P(\mathbf{l}_{i+1} | \mathbf{l}_i) P(\mathbf{q}_{i+1} | \mathbf{l}_{i+1}),
 \end{aligned} \tag{4}$$

where the last equality holds by assuming that

- (i) $P(\mathbf{q}_i | \mathbf{l}_i), i = 0, \dots, L$ are independent of each other, and
- (ii) $P(\mathbf{l}_0, \dots, \mathbf{l}_L)$ can be modelled by a first-order Markov chain. That is, $P(\mathbf{l}_i | \mathbf{l}_{i-1}, \mathbf{l}_{i-2}, \dots, \mathbf{l}_0) = P(\mathbf{l}_i | \mathbf{l}_{i-1})$ for all $i = 1, \dots, L$.

To evaluate (4), the transition probabilities between the feature lines \mathbf{l}_i and \mathbf{l}_{i+1} , $P(\mathbf{l}_{i+1} | \mathbf{l}_i)$, $i = 0, \dots, L - 1$, and the likelihoods $P(\mathbf{q}_i | \mathbf{l}_i)$, $i = 0, \dots, L$, have to be specified. First, we can see that $P(\mathbf{q}_i | \mathbf{l}_i)$ is a probability measure in association with the similarity between \mathbf{y}_i and \mathbf{q}_i , where \mathbf{y}_i is the projection point of \mathbf{q}_i onto \mathbf{l}_i . Hence, $P(\mathbf{q}_i | \mathbf{l}_i)$ can be set as being decreased with $\|\mathbf{q}_i - \mathbf{y}_i\|$, the distance between the observation \mathbf{q}_i to its projection point \mathbf{y}_i . We thus refer this probability to as the *probability caused by the distance from appearance manifolds* (PDAM). Second, because the image sequence to be recognized consists of consecutive views of an object, $P(\mathbf{l}_{i+1} | \mathbf{l}_i)$ is larger when \mathbf{l}_{i+1} is a more reasonable

consequent of \mathbf{l}_i by considering motion continuity. We refer $P(\mathbf{l}_{i+1}|\mathbf{l}_i)$ to as the *probability caused by motion continuity* (PMC). Nevertheless, note that here the concept of "motion continuity" has nothing to do with velocity or rotation, but rather only whether the relationship between images in the database matches the relationship between two images in the query.

3 Probabilistic Framework and Main Algorithm

3.1 Probability Distribution Setting for PDAM and PMC

To evaluate (4), we make the following assumptions:

First, the PDAM is modelled as a Gaussian distribution, $P(\mathbf{q}_k|\overline{\mathbf{x}}_u^c\overline{\mathbf{x}}_w^c) = \exp(-d_{c;u,w;k}^2/2\sigma^2)/Z$, where σ is a chosen constant, Z is a normalization constant which normalizes $P(\mathbf{q}_k|\overline{\mathbf{x}}_u^c\overline{\mathbf{x}}_w^c)$ to be a probability density function, and

$$d_{c;u,w;k} = d(\mathbf{q}_k, \overline{\mathbf{x}}_u^c\overline{\mathbf{x}}_w^c), \quad (5)$$

for $k = 0, \dots, L, c = 1, \dots, M, 1 \leq u, w \leq N_c$ and $u \neq w$.

Second, let the PMC be defined as follows:

$$P(\mathbf{l}_{i+1}|\mathbf{l}_i) = \begin{cases} 0; & \text{If } \mathbf{l}_{i+1} \text{ is not a neighbor of } \mathbf{l}_i, \\ \frac{1}{N(\mathbf{l}_i)}; & \text{Otherwise.} \end{cases} \quad (6)$$

where $N(\mathbf{l}_i) = 2N_c - 3$ is the number of neighboring FLs for \mathbf{l}_i .

Third, we assume equal priori probabilities for all the FLs by setting $P(\mathbf{l}) = 1/N_{total}$ for $\mathbf{l} \in \Gamma$.

Taking a natural log of (4), the following formulation can be derived:

$$\begin{aligned} \Psi^* &= \arg \max_{\Psi} \ln (P(\mathbf{l}_0)P(\mathbf{q}_0|\mathbf{l}_0) \prod_{i=0, \dots, L-1} P(\mathbf{l}_{i+1}|\mathbf{l}_i)P(\mathbf{q}_{i+1}|\mathbf{l}_{i+1})) \\ &= \arg \min_{\Psi} \ln J(\Psi; \Lambda_{\mathbf{q}}), \\ &= \arg \min_{\Psi} J(\Psi; \Lambda_{\mathbf{q}}), \end{aligned} \quad (7)$$

where

$$J(\Psi; \Lambda_{\mathbf{q}})^{-1} = \begin{cases} \infty & \text{if } P(\mathbf{l}_{i+1}|\mathbf{l}_i) = 0; \quad \text{for some } i \in \{0, \dots, L-1\}, \\ \sum_{i=0}^L d(\mathbf{q}_i, \mathbf{l}_i)^2/2\sigma^2 - \sum_{i=0}^{L-1} \ln(2N_c - 3) - \ln N_{total}; & \text{Otherwise.} \end{cases} \quad (8)$$

From (7), to find Ψ^* that maximizes the *a posteriori* probability is equivalent to find Ψ^* that minimizes the objective function J defined in (8). Note that it is computationally intractable to use brute force for computing Ψ^* . In this work, the PDAM and PMC are encoded in a matching graph, and dynamic programming is adopted for solving this minimization problem.

3.2 Construction of Matching Graph

We construct a graph G containing $L + 1$ levels. There are N_{total} nodes in each level, where N_{total} is the total number of feature lines defined before. For the k -th level, ($k = 0, \dots, L$), a number of N_{total} nodes, denoted by $n_{\overline{\mathbf{x}_u^c \mathbf{x}_w^c}; k}$, $c \in \{1, 2, \dots, M\}$ and $u, w \in \{1, 2, \dots, N_c\}$, are constructed in association with it. In addition to these nodes, we also construct a source node n_{-1} and a sink node n_{L+2} . Then, some edges are constructed by connecting nodes in G as follows. The source node n_{-1} and sink node n_{L+2} are fully connected to nodes at level 1 and level $L + 1$, respectively. For each adjacent levels k and $k + 1$, $k = 0, \dots, L - 1$, there is an edge $e(\overline{\mathbf{x}_{u_1}^c \mathbf{x}_{w_1}^c}; \overline{\mathbf{x}_{u_2}^c \mathbf{x}_{w_2}^c}; k)$ linking $n_{\overline{\mathbf{x}_{u_1}^c \mathbf{x}_{w_1}^c}; k}$ to $n_{\overline{\mathbf{x}_{u_2}^c \mathbf{x}_{w_2}^c}; k+1}$ if $\overline{\mathbf{x}_{u_1}^c \mathbf{x}_{w_1}^c} \in \Psi_{c, u_2, w_2}$, the set of neighbor FLs of $\overline{\mathbf{x}_{u_2}^c \mathbf{x}_{w_2}^c}$. Figure 2 shows an illustration of the graph G in association with the case in which there are two objects and each object has four views.

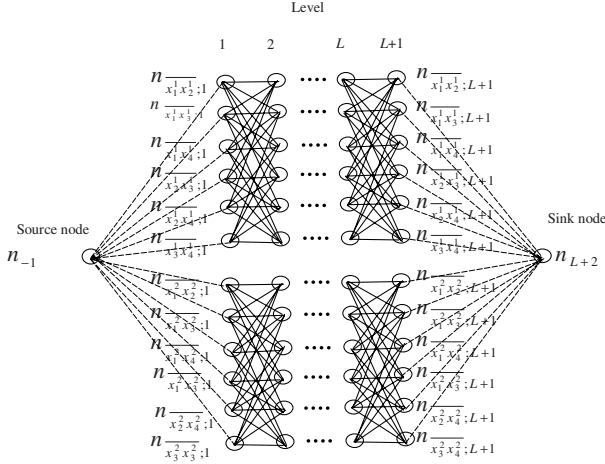


Fig. 2. An example of a graph G that is constructed for the case of two objects, where each object contains four views. There are therefore $6 = C_2^4$ FLs for each object

For each node $n_{\overline{\mathbf{x}_u^c \mathbf{x}_w^c}; k}$ (except the source and sink nodes), a node score $s_{c; u, w; k}$ is assigned to it by setting

$$s_{c; u, w; k} = d_{c; u, w; k}^2 / \sigma^2 \quad (9)$$

This node score is used to encode the log likelihood of the PDAM in (8). In addition, the scores of the source and sink nodes, s_{-1} and s_{L+2} , are both set to 0.

Then, the cost of a node is defined from the principle of dynamic programming as shown in the following. First, the cost of the source node, $cost(n_{-1})$, is set to zero. Then, the cost of each of the other nodes is defined recursively as

$$cost(n_{\overline{\mathbf{x}_u^c \mathbf{x}_w^c}; k}) = s_{c; u, w; k} + \min\{cost(n_{\overline{\mathbf{x}_{u'}^c \mathbf{x}_{w'}^c}; k'}) \mid n_{\overline{\mathbf{x}_u^c \mathbf{x}_w^c}; k'} \in \Theta_{c; u, w; k}\}; \quad (10)$$

for $k = 1, \dots, L + 2, c = 0, \dots, M$ and $0 \leq u, w \leq N_c$. Note that $\Theta_{c; u, w; k}$ is the set of nodes having edges linking to $n_{\overline{\mathbf{x}_u^c \mathbf{x}_w^c}; k}$.

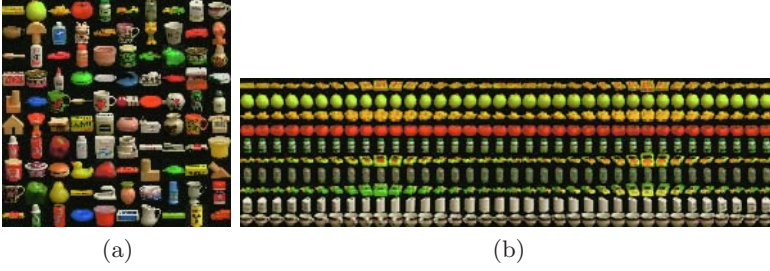


Fig. 3. Coil-100 data set. (a) The 100 objects in the Coil-100 data set. (b) Some views of ten of the objects shown in (a) (sampled by every 10°)

Each path starting from the source node and ending at the sink node represents a sequence of matches between a view belonging to $\Lambda_{\mathbf{q}}$ and a FL belonging to Γ . The cost of the sink node, $cost(n_{L+2})$, is then referred to as the minimal cost, and it is easy to verify that $cost(n_{L+2}) = \min_{\Psi} J(\Psi; \Lambda_{\mathbf{q}})$ defined in (8). The path associated with the minimal cost or, equivalently, the shortest path from the source to the sink nodes, is then referred to as the matching (or optimal) path in this work. The FL associated with nodes in G (except the source and sink nodes) are then treated as a sequence of matched FLs of the sequence of testing consecutive views, $\Lambda_{\mathbf{q}}$, and the object represented by these FLs then serves as the recognition result.

In our work, to avoid recursive programming, the Dijkstra algorithm [5] is used to find the optimal path. For each node, an incoming edge with the lowest accumulated cost is retained in our approach. After finding the best incoming choice for all nodes, our process backtracks, from the sink to the source nodes, to obtain the optimal path. Except for the source and sink nodes, each node passed by the optimal path then represents a match between a view belonging to $\Lambda_{\mathbf{q}}$ and a FL belonging to Γ .

4 Experimental Results and Discussions

4.1 Experimental Results

Coil-100 Object Recognition. The Coil-100 data set [11] was widely used as an object-recognition benchmark [10][14][15]. In this data set, there are 100 objects and each object has 72 different views (images) that are taken every 5° around an axis passing through the object. Each image is a 128x128 color one with R,G,B channels. Figures 3(a) and 3(b) show these 100 objects and some sampled views of ten of these objects, respectively.

We follow the experimental settings in [15], which used only a limited number of views per objects for training. In our experiment, four different views per object (0° , 90° , 180° and 270°) were used for training, as shown in Figure 4(a), and the other 6800 (i.e., $(72-4)*100=6800$) images were used for testing. In other words, for each object, 6 features lines can be constructed from its 4 training views.

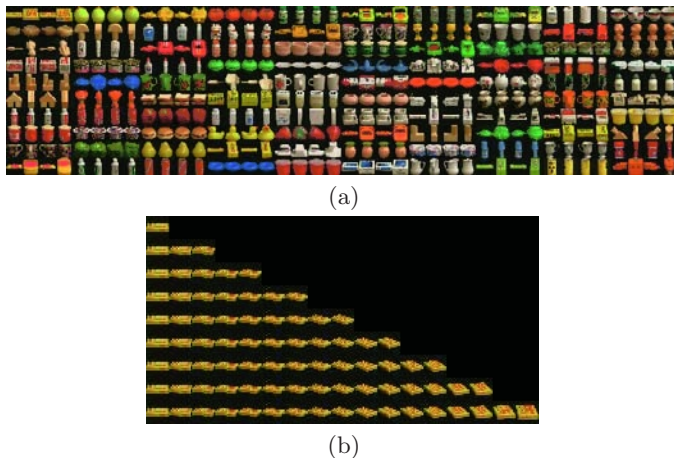


Fig. 4. (a) All the training views used in the experiments. (b) The sequences of views of the first object used in our experiment (with the lengths being 1 to 17)

In our experiment, the number of consecutive views (i.e., $L + 1$) used for testing was set to be 1, 3, 5, \dots , 17. A sequence of consecutive testing views of an object are shown in Figure 4(b). Note that when the number of consecutive testing views is 1, our method degenerates to the NFL method with only a single image as its input. For each number of consecutive testing views, the average recognition error rate over 6800 tests was computed and recorded. Note that the sets of training and testing views are disjoint in our experiment, and thus the training views were dropped out when an image sequence was sampled for testing.

In the first test, the input image format was set to be colored 16x16, which is the same as that in [14], and the 16x16 image directly serves as the feature vector. Figure 5 shows the experimental results by using our method. From this figure, it can be seen that our method can considerably improve the recognition performance of the NFL method as the number of consecutive testing views is increased. To further show that our probabilistic framework can integrate the consecutive visual clues better, two additional methods, NN-voting and NFL-voting, which are simply extended from the nearest-neighbor and nearest-feature-line methods to those employing image sequences by majority voting (i.e., the identity of recognition is determined by which receiving the maximal number of votes when every view in the sequence is independently recognized), respectively. As shown in Figure 5, our method outperform either NN-voting or NFL-voting because appearance-similarity and motion-continuity information is appropriately exploited.

In the second test, we investigate how the recognition error rates of our method can improve as more training images were added. In this test, 32x32 gray images from the same database were used, as those adopted in [15], and the 32x32 image directly serves as the feature vector. We vary the number of training views from 4 to 8 and Figure 6 shows these results. For example, when

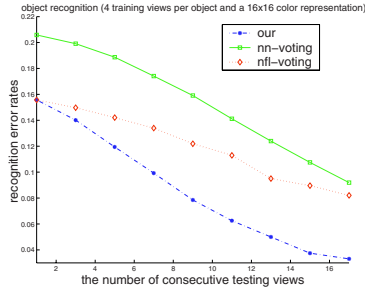


Fig. 5. Object recognition results using the same image format (i.e., color 16x16 images) as that in [14]. We compare our method (our) with nearest neighbor voting (nn-voting) and nearest feature line voting (nfl-voting) methods

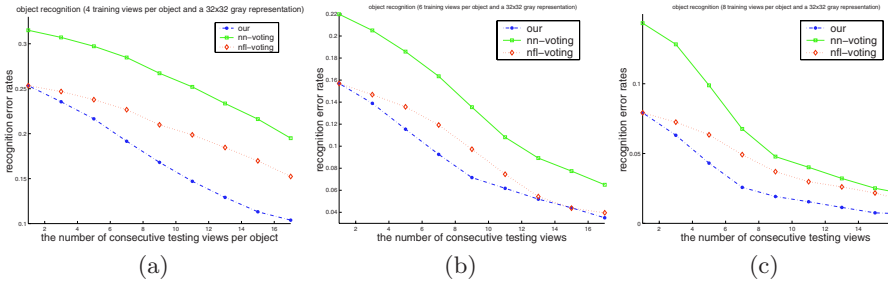


Fig. 6. Object recognition results using the same image format (i.e., gray level 32x32 images) as that in [15]. (a) 4 training views. (b) 6 training views (c) 8 training views

the numbers of training examples (FLs) increase from 4 (6) to 6 (15) and 8 (28), the lowest recognition error rates decrease from 10.38% to 3.5% and 0.67%, respectively. From these experimental results, it is shown that the performance of the method would dramatically improve as more images are added. This figure also shows that our method can greatly improve the single-view-based NFL method and are better than the NN-voting and NFL-voting methods.

Face Recognition. We perform face recognition on a face-only database ². There are 1280 image of 128 persons, where each person has 10 images with distinct poses or expressions per person. Each image size is normalized to be 32×32, which directly serves as the features being used. Figures 7 (a) and (b) show all 128 persons and all 10 views of the first 5 persons in this face database. In the following experiments, we use 5 different views (that is, 15 FLs) per person in training as shown in Figure 7(c) (left part), and other (i.e., (10-5)*128=640) images are used for testing. Hence, for each object, 15 features lines can be constructed from its 5 training views. In addition, the number of consecutive testing views is sampled from {1, 3, 5}. Some examples of sequences of three views are shown in Figure 7(c) (right part).

² This database can be download from <http://smart.iis.sinica.edu.tw/html/face.html>

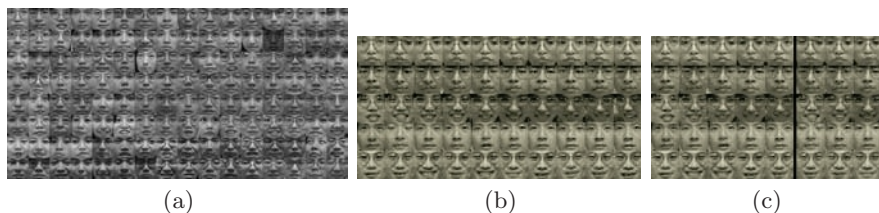


Fig. 7. A face-only database. (a) shows all 128 persons in this database. (b) shows all 10 views of the first 5 persons shown in the first row of (a). (c) shows training views (left part) and some examples of sequences of three consecutive testing views of these 5 persons (right part), respectively. Note that the training views are formed by collecting faces in columns 1, 3, 5, 7 and 9 shown in (b).

Table 1 shows the experimental results using our method and the comparisons between our method and NN-voting and NFL-voting. In addition, it also shows that our method give the best performance.

Table 1. Performance comparison between our method (our) with nearest neighbor voting (nn-voting) and nearest feature line voting (nfl-voting) methods on the face-only database. Recognition error rates in % on different numbers of consecutive testing views are shown. Our method gives the best performance.

	Ours	NN-Voting	NFL-Voting
number of testing views (1)	5.31	6.71	5.31
number of testing views (3)	0.26	2.86	2.6
number of testing views (5)	0	1.55	1.56

4.2 Discussions

To our best knowledge, no image-sequence-based methods have been used for testing these three databases in the past, and some existing results are introduced in the following for comparison.

For Coil-100 database, the following results have been reported. When the image format is colored 16x16, and the number of training views is four, the PCA-and-Spline-Manifold [10] and the linear SVM method [14] achieved 12.4% and 13.1% recognition error rates, respectively (the NFL method achieved 15.6% in our testing). When the image format is 32x32 gray, the linear SVM method [14] and the snow-with image method [15] achieve 18.4% and 21.5% error rates, respectively (the NFL method achieved 25.3% in our testing). From Figures 5 and 6(a), it can be seen that our method can outperform the existing results shown above when only 5 and 9 consecutive images of objects are used for colored 16x16 and 32x32 gray image formats (four training views), respectively.³ For the face-only face database, 13% recognition error rate can be achieved in [3] when

³ The above results were all tested based purely on image intensity information. Roth, Yang and Ahuja [15] have further exploited edge information to achieve a better

their method is tested on a part of this database (100 persons) and 6 images per person are used as training images. From Table 1, it can be seen that our method has lower recognition error rates even when only four training views are used. The results reveal that the recognition performance can be considerably upgraded by appropriately exploiting useful visual clues contained in an image sequence.

Some further remarks are addressed below:

Remark 1 [Neighborhood Relationships]: Although each of our experiments uses a database with a single linear sequence of views as the object rotates about a single axis, other kinds of neighborhood relationships in the database could be exploited. For example, our approach can also be used for the case that a database contains views as the object is rotated about two axes [6]. In this case, the neighborhood relationship is two dimensional but not one dimensional, and our approach is still applicable for this case.

Remark 2 [Feature Selection]: Although raw data was directly used as feature in our experiment, this is not the only choice. Our method can also use features produced by feature-extraction or feature-generation processes such as principal component analysis or linear discriminant analysis, and features generated in such ways would have chance to be helpful for either the computational efficiencies or the recognition accuracies.

5 Conclusions

There are several characteristics of our framework for appearance-based object recognition using a sequence of views. First, we emphasize inter-feature-line consistencies. Second, we take both the probability caused by the distance from manifold, PDAM, and the probability caused by motion continuity, PMC, into considerations. Third, to handle the associated recognition problem, we construct a matching graph in which PDAM and PMC are incorporated, and transform this problem into a shortest path problem that can be effectively solved by using dynamic programming. The experimental results on the Coil-100 data set and the face-only database show that our method achieves high recognition rates for object recognition and face recognition. Our method thus provides an effective way for appearance-based object recognition using a sequence of views.

Acknowledgments. This work is supported in part by the National Science Council of Taiwan under Grant NSC project, NSC 91-2213-E-001-022. J.-H. Chen acknowledges travel support from Department of Computer Science and Engineering, University of Washington at Seattle.

error rate, 11.7%, in their snow-with-image-and-edge method. Our result is better when the number of views in use is 15 without using edge information as shown in Figure 5 (four training views).

References

1. P. N. Belhumeur and D. J. Kriegman "What is the Set of Images of an Object under all Possible Illumination Conditions?," *International Journal of Computer Vision*, vol. 28, pp. 245–260, 1998.
2. M. J. Black and A. Jepson, "EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation," *International Journal of Computer Vision*, vol. 26, pp. 63–84, 1998.
3. L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu "A New LDA-based Face Recognition System Which Can Solve the Small Sample Size Problem," *Pattern Recognition*, vol. 33, pp. 1713–1726, 2000.
4. Y.-S. Chen, and et al. "Video-based Eye Tracking for Autostereoscopic Displays," *Optical Engineering*, vol. 40, pp. 2726–2734, 2001.
5. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.
6. Y.-P. Hung, C.-S. Chen, Y.-P. Tsai, S.-W. Lin, "Augmenting Panoramas with Object Movies by Generating Novel Views with Disparity-Based View Morphing," *Journal of Visualization and Computer Animation*, vol. 13, pp. 237–247, 2002.
7. S.Z. Li, K.L. Chan and C.L. Wang, "Performance Evaluation of the Nearest Feature Line Method in Image Classification and Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1335–1339, 2000.
8. S.Z. Li and J. Lu, "Face Recognition Using the Nearest Feature Line Method," *IEEE Transactions on Neural Networks*, vol. 10, pp. 439–443, 1999.
9. B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 696–710, 1997.
10. H. Murase and S. K. Nayar, "Visual Learning and Recognition of 3D Objects from Appearance," *International Journal of Computer Vision*, vol. 14, pp. 5–24, 1995.
11. S. A. Nene, S. K. Nayar and H. Murase, "Columbia Object Image Library (COIL-100)," *Technical Report CUCS-006-96*, Columbia University, 1996.
12. K. Ohba and K. Ikeuchi, "Detectability, Uniqueness, and Reliability of Eigen Windows for Stable Verification of Partially Occluded Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 1043–1048, 1997.
13. C. Papageorgious and T. Poggio, "A Pattern Classification Approach to Dynamic Object Detection," in *Proceedings of International Conference on Computer Vision*, Corfu, Greece, 1999, pp. 1223–1228.
14. D. Roobaert and M. M. van Hulle, "View-based 3D Object Recognition with Support Vector Machines," in *Proceedings of 1999 IEEE International Workshop on Neural Networks for Signal Processing*, pp. 77–84, Madison, Wisconsin, USA, 1999.
15. D. Roth, M.-H. Yang and N. Ahuja, "Learning to Recognize 3D Objects," *Neural Computation*, vol. 14, pp. 1071–1103, 2002.
16. S. Ullman and R. Basri, "Recognition by Linear Combinations of Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 992–1006, 1991.
17. T. Vetter and T. Poggio, "Linear Object Classes and Image Synthesis From a Single Example Image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 696–710, 1997.
18. P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 511–518, Kauai, Hawaii, USA, 2001.

19. C. Yuan and H. Niemann, "An Appearance Based Neural Image Processing for 3-D Object Recognition," in *Proceedings of International Conference on Image Processing*, Vancouver, Canada, pp. 344–347, 2000.
20. Z. Zhou, S. Z. Li and K. L. Chan, "A Theoretical Justification of Nearest Feature Line Method," in *Proceedings of International Conference on Pattern Recognition*, Barcelona, Spain, pp. 759–762, 2000.

Discriminant Analysis on Embedded Manifold*

Shuicheng Yan², Hongjiang Zhang¹, Yuxiao Hu¹, Benyu Zhang¹,
and Qiansheng Cheng²

¹ Microsoft Research Asia, Beijing, P.R. China

² School of Mathematical Sciences, Peking University, Beijing 100871, P.R. China

Abstract. Previous manifold learning algorithms mainly focus on uncovering the low dimensional geometry structure from a set of samples that lie on or nearly on a manifold in an unsupervised manner. However, the representations from unsupervised learning are not always optimal in discriminating capability. In this paper, a novel algorithm is introduced to conduct discriminant analysis in term of the embedded manifold structure. We propose a novel clustering algorithm, called Intra-Cluster Balanced K-Means (ICBKM), which ensures that there are balanced samples for the classes in a cluster; and the local discriminative features for all clusters are simultaneously calculated by following the global Fisher criterion. Compared to the traditional linear/kernel discriminant analysis algorithms, ours has the following characteristics: 1) it is approximately a locally linear yet globally nonlinear discriminant analyzer; 2) it can be considered a special Kernel-DA with geometry-adaptive-kernel, in contrast to traditional KDA whose kernel is independent to the samples; and 3) its computation and memory cost are reduced a great deal compared to traditional KDA, especially for the cases with large number of samples. It does not need to store the original samples for computing the low dimensional representation for new data. The evaluation on toy problem shows that it is effective in deriving discriminative representations for the problem with nonlinear classification hyperplane. When applied to the face recognition problem, it is shown that, compared with LDA and traditional KDA on YALE and PIE databases, the proposed algorithm significantly outperforms LDA and Mixture LDA, has better accuracy than Kernel-DA with Gaussian Kernel.

1 Introduction

Previous works on manifold learning [2][6][7][9] focus on uncovering the compact, low dimensional representations of the observed high dimensional unorganized data that lie on or nearly on a manifold in an unsupervised manner. These algorithms can be divided into two classes: 1) algorithms with mapping function only for sample data. The sample points are represented in a low dimensional space by preserving the local or global properties of a manifold, like ISOMAP [16], LLE [12], Laplacian Eigenmap [1]; 2) algorithms with mapping function for the whole data space. Roweis [13] presented an algorithm that automatically aligns a mixture of local dimensionality reducers into a single global representation of the data throughout

* This work was performed at Microsoft Research Asia.

space; Brand [3] presented a similar work to merge local representations and construct a global nonlinear mapping function for the whole data space. He [8] proposed the simple locality preserving projections to approximate the Laplacian Eigenmap algorithm. All these algorithms are unsupervised and most of them are only evaluated on toy problems.

In this paper, we propose an algorithm to utilize the class information for discriminant analysis in term of the manifold structure and applications in general classification problems such as face recognition. It is motivated by the following observations: First, previous works on manifold learning focus on exploring the optimal low dimensional representations that best preserve some characteristics of a manifold, while the best representative features are not always the best discriminant features for general classification task. On the other hand, the meaningful information may be lost in the dimensionality reduction, which in turn will degrade the posterior discriminant analysis based on the low dimensional data. Second, Linear Discriminant Analysis can only handle the linear classification problem and Kernel Discriminant Analysis [10] suffers from its heavy computation and memory cost although it can handle nonlinear cases in principle. The proposed algorithm is an efficient, low time and memory cost one for discriminant analysis based on the manifold structure.

For a curved manifold, the globally linearly inseparable manifold may be easily separable locally. The intuition of this work is to place some local Linear Discriminant Analyzers on a curved manifold, then merge these local analyzers into a global discriminant analyzer via global Fisher criterion. In the first step, the traditional methods such as Mixture Factor Analysis (MFA) [1] can not be directly applied, since they can not guarantee that there are balanced samples for the classes in a cluster and it's impossible to conduct Local discriminant analysis with only one class of samples in a cluster. In this work, we formulate this task as a special clustering problem and propose a novel clustering approach, called Intra-Cluster Balanced K-Means (ICBKM), to ensure that there are balanced samples for the classes in a cluster.

Taking the advantage of the clustering results of ICBKM, the sample data are reset as clusters, and local discriminant analysis can be conducted in each cluster. The traditional way to recognize a new data using these local analyzers is to conduct the classification using the nearest local analyzer. In this work, the local analyzers are dependent in both learning and inferring stage, and the optimal discriminative features for each cluster are computed simultaneously. First, PCA is conducted in each cluster; then the posterior probability of each cluster for a given data, i.e. $p(c|x)$ can be obtained. The optimal discriminative features for each cluster are computed by maximizing the global Fisher criterion, i.e. maximizing the ratio of the weighted global inter- and intra- scatters, where the scatters are computed based on the $p(c|x)$ weighted representations for the samples. In the inferring stage, the low dimensional representation for new data is derived as the $p(c|x)$ weighted sum of the projections from different clusters and the classification can be conducted using Nearest Neighbor (NN) algorithm based on the low dimensional representations. This algorithm can be justified in two different perspectives: 1) it automatically merges the local linear discriminant analyzers; and 2) it can be considered as a special kernel discriminant analysis algorithm with geometry-adaptive-kernel, in contrast to traditional KDA whose kernel is independent to the samples.

The rest of the paper is structured as follows. The intra-cluster balanced K-Means clustering method and global discriminant analysis based on the clustering results are introduced in section 2. In section 3, we present our justifications for the proposed algorithm in two different perspectives. The toy problem and the face recognition experiments compared with traditional LDA and KDA on YALE and PIE database are illustrated in section 4. Finally, we give the conclusion remarks in section 5.

2 Discriminant Analysis on Embedded Manifold

Suppose $X = \{x_1, x_2, \dots, x_N\}$ be a set of sample points that lie on or nearly on a low dimensional manifold embedded in a high dimensional observed space. For each sample $x_i \in \mathbb{R}^D$, a class label is given as $l_i \in \{1, 2, \dots, L\}$. Previous works on manifold learning are unsupervised and mainly focus on finding the optimal low dimensional embedding, i.e. the best low dimensional representations that preserve some characteristics of a manifold. However, the class information is not efficiently utilized in these algorithms and the derived representations are not always optimal for general classification task.

Here we show how to utilize the class information to conduct nonlinear discriminant analysis in term of the manifold structure. A continuous manifold can be considered as a combination set of a series of open sets; and for the discrete sample data on it, they can be considered as the combination of a series of clusters. On the other hand, the globally linearly inseparable manifold may be easily separable on these local open sets. It motivates us to conduct local discriminant analysis in these local clusters, and then merge these local analyzers into a global discriminant analyzer. Following this idea, we first segment the sample data into clusters. The traditional clustering algorithms like K-means [11] and Normalized Cut [14] can not be applied to the problem we concern here since there may be only single class in some clusters, which makes the local discriminant analysis impossible. To address this, we have proposed a clustering algorithm called *Intra-Cluster Balanced K-Means* to ensure that the sample numbers for the classes in a cluster are balanced. Secondly, we search for the local optimal features in each cluster by following the global Fisher Criterion in which we maximize the ratio of the cluster weighted inter- and intra-class scatters. In the following subsections, we will introduce the two steps of our algorithm in detail, respectively.

2.1 Intra-cluster Balanced K-Means Clustering

K-means clustering algorithm aims at putting the more similar samples in the same cluster. It is unsupervised, thus it can not guarantee that there is a balanced number of samples for the classes in a cluster. Compared to the general clustering algorithms, the clustering problem we concern here has some special characteristics: 1) the class label for each sample is presented and it can be supervised; 2) its purpose is not only to put the similar samples in the same cluster, but also to ensure that the samples for the classes in each cluster should be balanced since the local discriminant analysis will be conducted in each cluster. Cheung [4] proposed a variation K-Means approach called

ICBKM: Given the class label set $S_l = \{1, 2, \dots, L\}$, the data set \mathcal{X} , the class label l_i for each sample x_i in \mathcal{X} and the cluster number K .

1. **Initialization:** Compute the standard deviation \mathcal{D} of the data set \mathcal{X} ; randomly select $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_K$ as the initial cluster centers, then assign each x_i to the cluster whose center is the nearest to x_i ;
2. **Reset Cluster Centers:** For each cluster C^k , reset the center as the average of all the samples assigned to cluster C^k ;
3. **Assignment Optimization:** For each $x_i \in \mathcal{X}$, assign it to the cluster that makes the objective function smallest and the result satisfies the constraint in (1).
4. **Exchange Optimization:** For each cluster, exchange the cluster labels for the sample in C^k that is the farthest to cluster center and the sample $\notin C^k$ that is the nearest to cluster center. If no improvement, keep the previous labels.
5. **Evaluation:** If current step has no improvement, return the final clustering results $\{C^1, C^2, \dots, C^K\}$; else, go step 2;

Fig. 1. Intra-Cluster Balanced K-Means Algorithm

Cluster Balanced K-Means (CBKM), in which the concept cluster balance was proposed. However, it only ensures that the sample number in each cluster is balanced and does not take into account the class label information. To provide a solution to the special clustering problem, we propose a novel clustering approach named *Intra-Cluster Balanced K-Means* (ICBKM). ICBKM satisfies the requirement that there are balanced samples for classes in each cluster by adding an extra regularization term to constrain the sample number variation for the classes in each cluster.

Formally, the objective function of ICBKM can be represented as:

$$\arg \min_{K_i \in \{1, 2, \dots, K\}} \sum_i^N \frac{|x_i - \bar{x}^{K_i}|^2}{\mathcal{D}^2} + \alpha \sum_{k=1}^K |N^k - \bar{N}|^2 + \beta \sum_{k=1}^K \sum_{c=1}^{c_k} |N_c^k - \bar{N}^k|^2 \quad (1)$$

$$\text{subject to: } c_k \geq 2 \quad (k = 1, 2, \dots, K)$$

where \bar{x}^k is the average of the samples in cluster k ; N^k is the sample number in cluster k ; \bar{N} is the average sample number for each cluster; N_c^k is the sample number of the c -th class in cluster k ; \bar{N}^k is the average sample number for each class in cluster k ; c_k is the class number in cluster k ; α and β are the weighting coefficients for the last two terms.

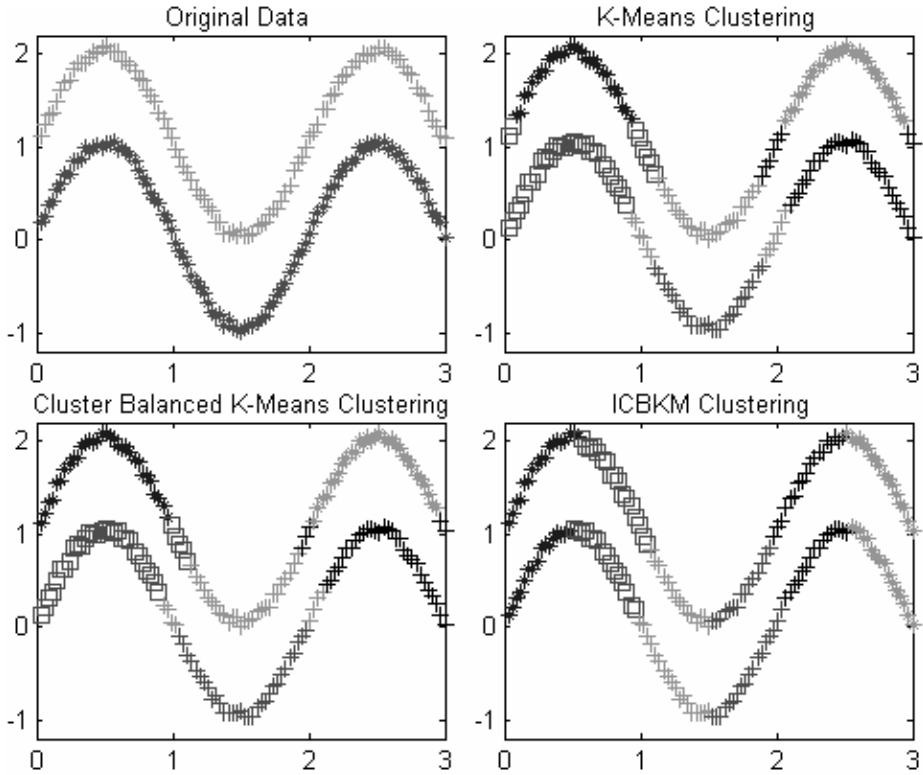


Fig. 2. Toy problem on synthesized data ($\alpha = 0.15, \beta = 0.1$)

In the objective function, minimizing the third term is to ensure that the classes have similar number of samples in a given cluster and the first two terms are the same as in CBKM. The objective function is not trivial and we can not obtain the close form solution directly. Here, we apply an iterative approach as traditional K-Means does. The Pseudo-code is listed in Figure 1. We present a new step for optimization called *Exchange Optimization* in ICBKM, in which the first term can be optimized while the last two terms are kept constant and the optimization conflict between the first term and last two terms that the assignment optimization step may face can be avoided.

The comparison experiments on the synthesized data are conducted and Figure 2 shows the results. It demonstrates that ICBKM presents intra-cluster balanced clustering results; and there is only one class in some clusters in the clustering results of K-Means and CBKM, which makes the local discriminant analysis impossible.

2.2 Global Discriminant Analysis by Merging Local Analyzers

Taking the advantage of the proposed Intra-Cluster Balanced K-Means approach, the sample data are segmented into clusters with balanced samples for different class. The

traditional way to utilize these clustering results is to conduct discriminant analysis in each cluster, then determine the class label for new data according to its nearest discriminant analyzer. In this way, the local analyzers is independent and the final classification use only part of the available information. We propose to utilize the global Fisher criterion to combine the local discriminant analyzers into a globally nonlinear discriminant analyzer. The global Fisher criterion maximizes the ratio of the weighed inter- and intra- cluster scatters. The entire algorithm has three steps and they are introduced in detail as follows:

PCA Projections: In each cluster, Principal Components Analysis (PCA) is conducted for dimensionality reduction; moreover, like in Fisher-faces, PCA step can prevent the algorithm from suffering from the singular problem when the sample number is less than the feature number. In all our experiments, we retain 98% of the energy in term of reconstruction error. Thus, in each cluster, each data $x_i \in \mathcal{X}$ can be presented as a low dimensionality feature vector:

$$z_i^k = (W_{pca}^k)^T (x_i - \bar{x}^k) \quad k = 1, \dots, K \quad (2)$$

in which W_{pca}^k is the leading eigenvectors. The conditional probability for cluster k given the data x , $p(C^k | x)$, can be obtained using a simple formulation [13]:

$$p(C^k | x) = p^k(x) / \sum_{j=1}^K p^j(x) \quad (3)$$

where $p^k(x) = \exp\{-\alpha^k(x)\}$ and $\alpha^k(x)$ is the *activity signal* of the data for cluster k . In our experiments, $\alpha^k(x)$ is set as the Mahalanobis Distance of the data in the PCA space of cluster k .

Nonlinear Dimensionality Reduction by Following Global Fisher Criterion: as previously mentioned, the linear discriminant analysis can not handle the nonlinear classification problem; and the KDA suffers from the heavy computation and memory cost in the classification stage. Here, we propose a novel discriminant analysis algorithm to conduct nonlinear discriminant analysis while need not to store samples for the feature extraction of the new data. The optimal features for all the clusters are simultaneously computed in a closed form and the local discriminant analyzers are automatically merged into a globally nonlinear discriminant analyzer by following the global Fisher criterion. For each sample $x_i \in \mathcal{X}$, it can be represented in cluster k as a low dimensional vector z_i^k . The purpose of the algorithm is to find the optimal feature directions w_f^k and the translations w_0^k for each cluster that minimizes the global Fisher criterion. Let $w^k = ((w_f^k)^T, (w_0^k)^T)^T$, the optimal representation for x_i is a weighted sum of the projections from different cluster:

$$\begin{aligned}
\Gamma(x_i) &= \sum_{k=1}^K P(C^k | x_i) (W_{pca}^k)^T (x_i - \bar{x}^k) \cdot w_f^k + w_0^k \\
&= \sum_{k=1}^K P_i^k (z_i^k \cdot w_f^k + w_0^k) = z_i \cdot w
\end{aligned} \tag{4}$$

where $w^T = ((w^1)^T, (w^2)^T, \dots, (w^K)^T)$ and $z_i^T = ((p_i^1 z_i^1)^T, p_i^1, \dots, (p_i^K z_i^K)^T, p_i^K)$. And the global intra-class and inter-class scatter can be represented as:

$$S_w = \sum_{i=1}^N (\Gamma(x_i) - \bar{\Gamma}^l) (\Gamma(x_i) - \bar{\Gamma}^l)^T = w^T \sum_{i=1}^N (z_i - \bar{z}^l) (z_i - \bar{z}^l)^T w = w^T M_w w \tag{5}$$

$$S_b = \sum_{l=1}^L N_l (\bar{\Gamma}^l - \bar{\Gamma}) (\bar{\Gamma}^l - \bar{\Gamma})^T = w^T \sum_{l=1}^L N_l (\bar{z}^l - \bar{z}) (\bar{z}^l - \bar{z})^T w = w^T M_b w \tag{6}$$

where $\bar{\Gamma}^l$ is the mean of $\Gamma(x)$ belonging to class l and $\bar{\Gamma} = \frac{1}{N} \sum_{i=1}^N \Gamma(x_i)$. The global

Fisher criterion is to maximize the cluster weighted inter-class scatter while minimize the cluster weighted intra-class scatter, *i.e.*

$$w^* = \arg \max_w \frac{|w^T M_b w|}{|w^T M_w w|} \tag{7}$$

It has close form solution and can be directly computed out using generalized eigen-decomposition algorithm [5].

Nonlinear Dimensionality Reduction for Classification: For a new data, the posterior probabilities for each cluster can be computed according to Eqn (3) and its low dimensional representation is obtained via the following nonlinear mapping functions in term of the derived local features in each cluster:

$$M(x) = \sum_{k=1}^K P(C^k | x) (W_{pca}^k)^T (x - \bar{x}^k) \cdot w_f^{*k} + w_0^{*k} \tag{8}$$

It is an explicit nonlinear mapping function from the data space to the low dimensional space. The consequent classification can be conducted based on these low dimensional representations using the traditional approaches like Nearest Neighbor (NN) or Nearest Feature Line (NFL). In all our experiments, we used the NN for final classification.

3 Justifications

Our proposed algorithm for discriminant analysis on embedded manifold (Daemon) consists of two steps: 1) separate the samples into class balanced clusters; and 2) merge the local discriminant analyzers into a global nonlinear discriminant analyzer by following the global Fisher criterion. It supervises the local analyzers and automatically decides the responsibility for each analyzer, which is somewhat like the background procedure named Daemon in UNIX system, thus this algorithm is referred as Daemon in the following. The intuition of Daemon is to merge the local discriminant analyzers into a unified framework; while it can be understood from a different perspective: it is a special kind of Kernel Discriminant Analysis algorithm, in which the kernel is data adaptive and geometry dependent; unlike other kernel machines that are independent to the samples they will analysis. In the following, we will discuss these two points in detail.

Automatically Merging Local Discriminant Analyzers: In the first step, Daemon uses ICBKM for clustering and ICBKM ensures that the derived cluster has balanced samples for the classes, thus local discriminant analysis can be conducted in each cluster. Daemon merges these local discriminant analyzers by following global Fisher criterion. The local optimal directions in each cluster are dependent and computed out simultaneously, which is different from the traditional way to utilize clustering results in which local analyzers are independent. In the classification stage, these local analyzers are also dependent and the final representation is a weighted sum of the outputs from these local analyzers. The Eqn (8) can be also presented as:

$$M(x) = \sum_{k=1}^K P(C^k | x) FE_k(x) \quad (9)$$

in which $FE_k(x)$ is the feature extractor in cluster k . As shown in Fig 3, the local analyzer has different optimal feature direction and it is locally discriminative; moreover, they can be merged and result in a globally nonlinear discriminant analyzer.

Special Kernel Discriminant Analysis: Daemon follows the global Fisher criterion in the learning stage and intrinsically is a discriminant analysis algorithm; on the other hand, it is a special kind of Kernel Discriminant Analysis algorithm with geometry-adaptive-kernel. The traditional kernel machine is manually defined and independent to the sample data. As shown in Eqn (4), Daemon can be considered a process in which the training data is mapped into another data space $\{z\}$, and then LDA are conducted in the new feature space. Therefore, Daemon can be considered a special Kernel Discriminant Analysis algorithm and the kernel is:

$$k(x, y) = \phi(x) \cdot \phi(y) \quad (10)$$

where $\phi(x) = z(x) = ((p^1 z^1)^T, p^1, \dots, (p^K z^K)^T, p^K)^T$ in which $z(x)$ is defined as in Eqn (4). The kernel has the following characteristics: 1) it has explicit mapping function from the input space to another feature space as the polynomial kernel does;

and 2) the kernel is dependent on the training samples and adaptive to the geometry structure. It can be solved just like a traditional Kernel Discriminant Analysis algorithm. Let the sample matrix $X_k = (\varphi(x_1), \varphi(x_2), \dots, \varphi(x_N))$ and the optimal feature direction $\varphi = X_k v$, the problem is changed as follows:

$$w^* = \arg \max_w \frac{|v^T K M K v|}{|v^T K N K v|} \quad (11)$$

where $K_{ij} = k(x_i, x_j)$, $M = \sum_{l=1}^L P_l - P$, $N = I - \sum_{l=1}^L P_l$; and

$$P_l = \frac{1}{N_l} e_l e_l^T, \quad P = \frac{1}{N} e e^T, \quad e \text{ is a vector with ones, } e_j(i) = \delta_{i,j}.$$

It can be solved using generalized eigen-decomposition algorithm and the projection of a new data onto the discriminant direction is:

$$M_k(x) = (\varphi, \phi(x)) = \sum_{i=1}^N v_i k(x_i, x) \quad (12)$$

It's obvious that $X_k v = w$ since Eqn (7) is equal to Eqn (11) when w is replaced by $X_k v$. Consequently, $M(x)$ in Eqn (8) and $M_k(x)$ in Eqn (12) are also equal. In other words, daemon is a Kernel Discriminant Analysis algorithm with explicit nonlinear mapping function from the input space to another feature space; moreover, as it is designed to be adaptive to the special data geometry structure, it should have strong ability to cope with the nonlinearly distributed data.

4 Experiments

In this section, we present two types of experiments to evaluate the Daemon algorithm. The experiment on toy problem of the synthesized data demonstrates the effective of ICBKM to derive discriminative feature in nonlinear classification problem; and the face recognition results on YALE and CMU PIE database shows that Daemon significantly outperforms LDA and has slightly better accuracy than traditional KDA.

4.1 Toy Problem

As shown in the upper-left image of Figure 2, the original data is composed of two classes of samples and they can not be separated linearly. They are synthesized according to the following function:

$$\begin{cases} x_i^k = 0.03 * i + \delta \\ y_i^k = \sin(\pi x_i) + k + \delta \end{cases} \quad k=0, 1, \delta \sim N(0,0.1) \quad (13)$$

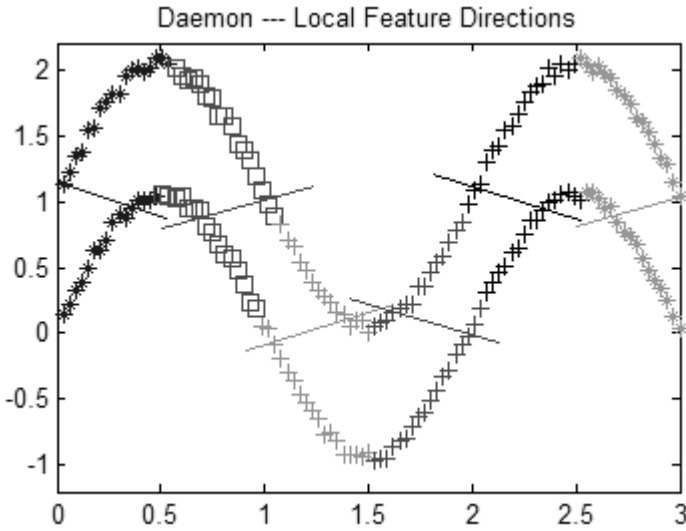


Fig. 3. The derived local feature directions using Daemon

We have systematically compared the clustering results of three K-Means-like algorithms. The original K-Means algorithm produced clustering results that aim at least sums of intra-cluster variances. As shown in the up-right image of Figure 2, the sample numbers for the classes in a cluster is not balanced and some clusters have only one class of samples. It makes the consequent local discriminant analysis impossible in these clusters. The cluster-balanced K-means algorithm produced similar result as that of the original K-means algorithm, yet, the sample number for each cluster is balanced.

As shown in the down-right image of Figure 2, the clustering result from our proposed ICBKM algorithm has the following properties: 1) the sample numbers for the classes in a cluster are balanced, which fascinates the local discriminant analysis in each cluster; and 2) the two classes of samples in each cluster can be easily separated. It is obvious that our proposed ICBKM algorithm produced more useful clustering result than the other two methods and Intra-cluster balanced clustering result presents proper structure representation for the following analysis. The computed local feature direction in each cluster is illustrated in Figure 3. It shows that the local feature direction is approximately optimal for the samples in a cluster.

4.2 Face Recognition

In this subsection, the YALE [17] and PIE [15] databases are used for face recognition experiment. In both experiments, the face image is normalized by fixing the eyes in the same position and each pose uses a different position. Yale face database is constructed at the Yale Center for Computational Vision and Control. It contains 165 grayscale images of 15 individuals. For each individual, six faces are used for training, and the other five are used for testing. Table 1 illustrated the face

Table 1. Comparison between LDAs, Kernel-DA and Daemon on Yale database

Algorithm	Fisher-faces	Kernel-DA	Mixture-LDA	Daemon(K=2)
Accuracy	80%	84%	82.7%	88%

Table 2. Comparison between Fisher-faces, Kernel-DA and Daemon on PIE database

Algorithm	Fisher-faces	Kernel-DA	Daemon(K=5)
Accuracy(67 Dim)	63.63%	67.79%	71.12%

recognition results of the Daemon and LDA, KDA and Mixture LDA that trains different LDA model for each cluster from ICBKM. It shows that Daemon significant outperforms LDA and Mixture LDA, has better results than traditional KDA with Gaussian Kernel.

We have also conducted the multi-view face recognition on the PIE database. We used the face images of pose 02, 37, 05, 27, 29 and 11 with out-plane view variation from -45° to 45° in our experiments. We averaged the results over 10 random splits. The experimental results illustrated in Table 2 again show that Daemon outperforms the other two algorithms. It is demonstrated again that Daemon has strong capability to handle nonlinear classification problems and can improve the accuracy in the general classification problems compared with LDA.

5 Discussions and Future Directions

We have presented a novel algorithm called Daemon for general nonlinear classification problem. Daemon is a nonlinear discriminant analysis algorithm in term of the embedded manifold structure. In this work, the discrete sample data on a manifold is clustered by our proposed Intra-Cluster Balanced K-Means algorithm such that the sample numbers for the classes in a cluster are balanced; and then the local optimal discriminant features are simultaneously derived by following the global Fisher Criterion. It is solved via general Eigen-decomposition algorithm. Daemon can be justified as an automatic merger of the local discriminant analyzers by following the global Fisher criterion; and it can be also justified as a special kernel discriminant analysis algorithm with geometry-adaptive-kernel.

To the best of our knowledge, it is the first work to conduct discriminant analysis while explicitly considering the embedded geometry structure. In this work, we have only utilized the basic property of manifold that a manifold can be covered by a series of open sets; how to combine the other topology properties of a manifold with discriminant analysis for general classification problem is the future direction of our work, and we are considering it in theory and applications.

References

- [1] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering", *Advances in Neural Information Processing System* 15, Vancouver, British Columbia, Canada, 2001.
- [2] C. Bregler, S.M. Omohundro, "Nonlinear manifold learning for visual speech recognition," *Fifth International Conference on Computer Vision*, June 20-23, 1995.
- [3] M. Brand, "Charting a manifold", *Advances in Neural Information Processing Systems* 15, 2002.
- [4] D.W. Cheung, S.D. Lee and Y. Xiao, "Effect of Data Skewness and Workload Balance in Parallel Data Mining". *IEEE Transaction on Knowledge and Data Engineering*, V.14 N.3, pp. 498–513, May 2002.
- [5] F.R. K. Chung, "Spectral Graph Theory", *Regional Conferences Series in Mathematics*, number 92, 1997.
- [6] D. Freedman. "Efficient simplicial reconstructions of manifolds from their samples", in *IEEE Trans. On PAM*, Vol: 24 Issue: 10 , Oct 2002, Page(s): 1349–1357.
- [7] J. Gomes, and A. Mojsilovic, "A variational approach to recovering a manifold from sample points", *Proc. European Conf. Computer Vision*, Copenhagen, May 2002.
- [8] Xiaofei He, Partha Niyogi. "Locality Preserving Projections (LPP)". TR-2002-09, 2002.
- [9] G. Hinton and S. T. Roweis. Stochastic Neighbor Embedding. *Advances in Neural Information Processing Systems* 15, 2002.
- [10] Qingshan Liu, Rui Huang, Hanqing Lu, Songde Ma. "Face Recognition Using Kernel Based Fisher Discriminant Analysis", in *Proceeding of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition* Page 197: May 20-21, 2002.
- [11] J. MacQueen. "On convergence of k-means and partitions with minimum average variance", *Ann. Math. Statist.*, 36:, 1965.
- [12] S. T. Roweis, and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding", *Science*, vol 290, 22 December 2000.
- [13] S. Roweis, L. Saul and G. Hinton, "Global Coordination of Local Linear Models", *Advances in Neural Information Processing System* 14, 2001.
- [14] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22 (2000), 888–905.
- [15] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression (PIE) Database", in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, May, 2002.
- [16] J. B. Tenenbaum, Vin de Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction", *Science*, vol 290, 22 December 2000.
- [17] Yale Univ. Face Database, <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>, 2002.

Multiscale Inverse Compositional Alignment for Subdivision Surface Maps

Igor Guskov

University of Michigan, Ann Arbor MI 48109, USA
guskov@eecs.umich.edu

Abstract. We propose an efficient alignment method for textured Doo-Sabin subdivision surface templates. A variation of the inverse compositional image alignment is derived by introducing smooth adjustments in the parametric space of the surface and relating them to the control point increments. The convergence properties of the proposed method are improved by a coarse-to-fine multiscale matching. The method is applied to real-time tracking of specially marked surfaces from a single camera view.

1 Introduction

Real-time tracking of textured surfaces in video is important for applications in user tracking and shape acquisition. In both of these applications active appearance models (AAMs) have been employed in a wide variety. A template matching procedure is often used as a basic component of AAMs. In order to accommodate the geometry of the tracked object the template is formulated as a textured shape. Both polygonal meshes and smooth splines were used as the underlying surface representations [1][2]. In this paper we develop a template tracking method for subdivision surfaces. Subdivision surfaces [3] offer a smooth and general representation of shape widely used in graphics and animation.

Inverse compositional template matching was recently proposed for the active appearance models based on triangular meshes [1][4]: it separately applies the current and incremental warps to the image and the template; the result is an efficient update procedure. For surface templates the separation of the incremental and current warps means that the incremental warp is performed in the parametric space of the surface. Thus, an additional difficulty arises on how to construct a space of such parametric warps. We propose a systematic approach to the construction of smooth atomic warps in the parameter space. Our derivations are first done for the ideal case of smooth warps, and then approximated in the space of subdivision surfaces.

The real-time operation of the template alignment procedure requires an efficient implementation. The natural multiresolution representation of subdivision surfaces serves as a suitable framework for implementing a coarse-to-fine matching algorithm that improves the convergence properties of our method. The multiscale approach also overcomes the deficiencies of the single resolution template matching for discontinuous textures.

Related work. Our work builds on the strengths of inverse compositional image alignment method developed by Baker and Matthews [4]. We extend their framework to handle templates specified on subdivision surfaces and propose a principled way of designing a set of smooth parametric adjustments (Section 3.3). The power of subdivision surfaces allow to represent big smooth patches of an arbitrary surface and control them with few control points. In this setting it is natural to consider coarse-to-fine version of the matching algorithm (Section 6). This improves the convergence properties of the matching without resorting to eigen-tracking approaches. Thus, it places less restrictions on the space of allowable surface deformations and does not require an apriori knowledge of the surface deformation model. The multiscale approach has been applied successfully for many related problems such as optical flow[5][6] and template search[7].

The goal of our work is similar to the Active Blobs effort [8]. While our method can handle general textured surfaces, it achieves its best tracking performance on specially quad-textured surfaces due to better control over conditioning of the involved matrix computations throughout all the levels of the hierarchy. In this paper, we only implement a simple appearance variation model and do not handle the detection of occluded regions and outlier pixels. Rather, the focus is on developing an efficient inverse compositional alignment procedure for general smooth surfaces described as subdivision models.

The paper is organized as follows: Section 2 introduces compositional template matching, Section 3 talks about surface and warp representations used in our work, Section 4 contains the detailed description of the proposed method. Sections 5 and 6 cover the partial template matching and the multiscale approach. Section 7 discusses the obtained results.

2 Compositional Template Matching

2.1 Forward Compositional Methods for Surfaces

The goal of template matching is to find the best warp of a template to match a given image. In the scenario of surface tracking, the template is better treated as a function on the parametric space of the tracked surface. Formally, let Ξ be the parametric space of the surface. Denote the projection of the surface in the image as $S(\xi)$, so that $S : \Xi \rightarrow \mathbf{R}^2$; we shall call the function S a *surface map*. The template function T represents surface color, so that $T : \Xi \rightarrow \mathcal{C}$, where $\mathcal{C} = \mathbf{R}$ for grayscale images and $\mathcal{C} = \mathbf{R}^3$ for color images.

Given an image $I : \mathbf{R}^2 \rightarrow \mathcal{C}$, the matching problem consists of finding the surface map S which minimizes the error functional

$$E(S) := \|I \circ S - T\|^2 = \int_{\Xi} (I(S(\xi)) - T(\xi))^2 d\xi.$$

At this point we assume that all of the surface is visible, the case of partial visibility is considered in Section 5. We would also like to delay specifying a particular representation for the surface map S and treat it as a general smooth function in this section.

The two approaches to the template matching problem are the *additive* and *compositional* methods. Additive approach performs update $S \leftarrow S + dS$, and finds the optimal surface adjustment dS by solving $\min_{dS} \|I \circ (S + dS) - T\|^2$. The compositional approach updates the surface map via $S \leftarrow S \circ W$. It looks for the optimal warping in the parametric space: $\min_W \|I \circ S \circ W - T\|^2$, or in more detail:

$$\min_W \int_{\Xi} (I(S(W(\xi))) - T(\xi))^2 d\xi. \quad (1)$$

The two approaches can be shown to be equivalent when the incremental warp $W : \Xi \rightarrow \Xi$ is close to the identity map so that $W(\xi) \approx \xi + dW(\xi)$ and dW is small. Then the corresponding surface adjustment will be close to

$$S \circ W - S \approx \frac{\partial S}{\partial \xi} dW. \quad (2)$$

Details of the proof can be found in [4]. When the Jacobian $\partial S / \partial \xi$ is not full rank the two approaches are not equivalent. This happens, for instance, in the silhouette region of the surface map where the 2D tangential space gets projected onto a 1D line in the image.

The compositional approach is possible in its pure form when one can find a set of planar warps which form a group. When representing a general evolving surface, one requires more flexibility than present in the classical groups of transformations such as translations, affine transforms or homographies. On the other hand, the very general group formed by composition of arbitrary smooth maps used above is not practical. Our approach will be therefore to derive compositional methods for the general smooth case, and then approximate the needed computations in a smooth basis.

2.2 Inverse Compositional Method

We shall now derive the *inverse* compositional method [4] in the general case of smooth surface maps and warps. The basic assumption for equivalence between the forward and inverse compositional methods is the closeness of the incremental warp map to being the identity map. In particular, it is assumed that $\det(\partial W / \partial \xi) \approx 1$. Then, the change of variable $\xi = W^{-1}(\eta)$ in (1) leads to the following minimization problem:

$$\min_W \int_{\Xi} (I(S(\eta)) - T(W^{-1}(\eta)))^2 d\eta.$$

The inverse of the incremental warp $V := W^{-1}$ can be sought directly:

$$\min_V \int_{\Xi} (I(S(\eta)) - T(V(\eta)))^2 d\eta.$$

This approach results in less per frame computation than in the corresponding forwards methods as was shown in [4]. Once the incremental warp is found, we can update the surface map via $S \leftarrow S \circ V^{-1}$.

3 Template and Warp Representation

The preceding section introduced the inverse compositional method in a general form. In practice, we need to use a specific representation for both the subdivision surface S and the parametric warp W . This section describes surface and warp representations used in our approach.

3.1 Subdivision Surface Maps

We define our template surface maps using Doo-Sabin subdivision scheme [9]. In this section, we give short description of our implementation. For a detailed introduction to subdivision surface modeling the reader is referred to [10] [3].

A subdivision surface is controlled by a control polygonal mesh \mathcal{M} that has a set of control vertices \mathcal{V} . For each control vertex $k \in \mathcal{V}$, a planar position $p_k \in \mathbf{R}^2$ is specified. The Doo-Sabin subdivision scheme is a *dual* scheme, so that its (dual) control vertices correspond to faces of a primal polygonal mesh \mathcal{M}' . We restrict the primal mesh to be a manifold quad mesh possibly with boundary.

The parametric space Ξ of a subdivision surface map is formed as the union of square patches Ξ_k ($k \in \mathcal{V}$) glued along the shared edges. Each patch is a square $[0, 1] \times [0, 1]$, and is associated to a particular primal face from \mathcal{Q}' or the corresponding dual control vertex from \mathcal{V} . A point ξ in a parametric space is then fully described by the (dual) control vertex index k and a position in $[0, 1] \times [0, 1]$. Given a function $f : \Xi \rightarrow \mathbf{R}^d$, we shall use notation $\partial f / \partial \xi^i, i = 1, 2$ for its derivatives. This is well defined within patches away from the patch boundaries, which is sufficient for the purposes of this paper.

We use primal-dual approach described in [11] to implement subdivision. The corner vertices are dependent on the boundary and inside vertices that share the same control face of the mesh; we also exclude the corner patches from the parametric region of the template. Thus, the corner control vertices only appear as an auxiliary dependent quantity; to simplify notation we redefine the set of control vertices \mathcal{V} as the union of inside and boundary vertices in the remainder of this paper, with the parametric region Ξ defined correspondingly to exclude the corner patches.

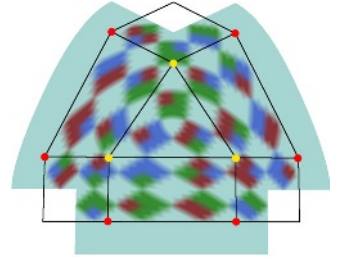
Having defined parametric region Ξ and the set of control vertices \mathcal{V} , we define the C^1 subdivision surface map at any parametric position ξ via

$$S[p](\xi) := p_k \phi^k(\xi),$$

where the summation in k is assumed over every index in \mathcal{V} , and p_k 's are two-dimensional points.

The following properties of $\phi^k(\xi)$ are important:

- $\sum_{k \in \mathcal{V}} p_k \phi^k(\xi) = 1$ for $\xi \in \Xi$, so that $\phi^k, k \in \mathcal{V}$ form the partition of unity.

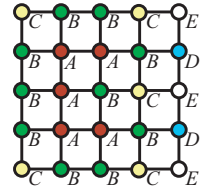


- Each $\phi^k(\xi)$ has local support in Ξ that consists of the patch of k together with all the patches that share at least one patch corner with it. Thus, each parametric point ξ is only affected by few control points whose support includes ξ .

3.2 Control Mesh Extension

In this section we describe a procedure for extrapolating the control mesh of the subdivision map from a given subset of (dual) control vertices with known positions to a wider set of control vertices adjacent to this known region. We shall use this procedure in two situations: creation of a canonical control position arrangement for the atomic warps as described in Section 3.3, and extension of active region positions for partial template matching as described in Section 5.

Let \mathcal{V}_A be the set of *active* control vertices, and define the set \mathcal{V}_B of non-active control vertices immediately adjacent to the set \mathcal{V}_A . Suppose that the positions of control vertices in both sets \mathcal{V}_A and \mathcal{V}_B are known. We introduce three sets of control vertices, depending on their relative adjacency to the known vertex set, and define a procedure for extending positions in $\mathcal{V}_A \cup \mathcal{V}_B$ to these three sets. The figure on the right shows an example of vertex set assignment.



- \mathcal{V}_C is the set of control vertices (not in \mathcal{V}_A or \mathcal{V}_B) whose primal faces share at least one primal control vertex with the primal control faces corresponding to control vertices in \mathcal{V}_A . In other words, a control vertex from \mathcal{V}_C will belong to at least one dual control face whose vertices include at least one vertex from \mathcal{V}_A . Note that the same dual face will also include at least two vertices from \mathcal{V}_B (this will be important for the extrapolation method described below).
- $\mathcal{V}_D := \text{bou}(\mathcal{V}_A \cup \mathcal{V}_B) \setminus (\mathcal{V}_A \cup \mathcal{V}_B \cup \mathcal{V}_C)$.¹ Each vertex in \mathcal{V}_D has at least one adjacent vertex $v_b \in \mathcal{V}_B$. By construction, the vertex v_o on the opposite side of v_b will also be from $\mathcal{V}_A \cup \mathcal{V}_B$.
- \mathcal{V}_E is the set of control vertices (not in $\mathcal{V}_A \cup \mathcal{V}_B \cup \mathcal{V}_C \cup \mathcal{V}_D$) whose primal faces share at least one primal control vertex with the primal control faces corresponding to control vertices in \mathcal{V}_B .

For a vertex in \mathcal{V}_C take a dual face that has at least three known vertices. Let this dual control face have n vertices, and index its corners with integers $i = 0, \dots, n-1$ in a counterclockwise order. We associate the vertex i with the parameter value $\alpha_i = 2\pi i/n$, and find the ellipse $p(\alpha) = C + D_1 \cos \alpha + D_2 \sin \alpha$ that best fit the known points in the least square sense. We then assign all the unknown vertex positions on the ellipse at the appropriate α locations. In the regular case of three known vertices and a single unknown vertex, we obtain a parallelogram rule. If there are several prediction for a vertex in \mathcal{V}_C we compute their average.

¹ For a subset of vertices $\mathcal{U} \subset \mathcal{V}$ we define its boundary $\text{bou}(\mathcal{U})$ as the set of vertices from $\mathcal{V} \setminus \mathcal{U}$ adjacent to at least one vertex in \mathcal{U} .

Now for each vertex in \mathcal{V}_D we extrapolate its value linearly via $p(v_d) = 2p(v_b) - p(v_o)$ where v_b and v_o are as described above. After this step, all the vertices in $\mathcal{V}_A \cup \mathcal{V}_B \cup \mathcal{V}_C \cup \mathcal{V}_D$ are assigned a position. We can now replicate the extrapolation step of \mathcal{V}_C for all the vertices in \mathcal{V}_E , which concludes our extrapolation procedure.

The described procedure guarantees the position assignment for all the control vertices that affect the subdivision map values within parametric patches associated with $\mathcal{V}_A \cup \mathcal{V}_B$. This will be especially important for the partial template matching of Section 5.

3.3 Warp Space

For a surface map represented as $S[p](\xi) = \phi^k(\xi)p_k$, the control point adjustments Δp result in the surface sample positions changed by $\phi^k(\xi)\Delta p_k$. For the compositional approach to work, this surface adjustment has to match the surface adjustment obtained via warping. Any smooth parameterized warp map $W(\xi; q)$ such that $W(\xi; 0) = \xi$ can be written:

$$W(\xi; q) = \xi + \frac{\partial W(\xi; 0)}{\partial q} q + O(q^2).$$

Define $\gamma_{i'}^{ki}(\xi) := \partial W^i(\xi; 0)/\partial q_k^{i'}$. For the purpose of template matching, these first derivatives are all that is necessary to define explicitly.

For a fixed index K , we choose γ^K in such a way that for some configuration of control points \bar{p}^K the surface map update corresponding to the parametric control point q_K matches the surface update coming from the control point adjustment Δp_K . More precisely, we would like that for all $\xi \in \Xi$:

$$\phi^k(\xi)\Delta p_k^i = \frac{\partial S^i[\bar{p}^K](\xi)}{\partial \xi^j} \gamma_{i'}^{kj}(\xi) q_k^{i'},$$

when $\Delta p_k^i = \delta_k^K Z^i$ and $q_k^{i'} = \delta_k^K Z^{i'}$ (here Z is an arbitrary two-dimensional vector). It follows that (no summation on the capital index K):

$$\phi^K(\xi)\delta_{i'}^i = \frac{\partial S^i[\bar{p}^K](\xi)}{\partial \xi^j} \gamma_{i'}^{Kj}(\xi).$$

The quantity on the left hand side is a scaled identity matrix, therefore we can conclude that $\gamma_{i'}^{Kj}(\xi)$ is the inverse of the 2×2 matrix $\partial S[\bar{p}^K](\xi)/\partial \xi$ times the scalar value $\phi^K(\xi)$ at each ξ . Hence we set

$$\gamma^K(\xi) := \left(\frac{\partial S[\bar{p}^K](\xi)}{\partial \xi} \right)^{-1} \phi^K(\xi). \quad (3)$$

Each $\gamma_{i'}^{Kj}(\xi)$ defined above is a local function, non-zero on the support of the basis function $\phi^k(\xi)$.

We now need to define \bar{p}^K (for this canonic arrangement of control points around a vertex K the warped surface adjustment will have an exact representation in the basis of subdivision shapes). Note that only evaluation of $S[\bar{p}^K]$ within the support of a single basis function $\phi^k(\xi)$ is required. We use the control mesh extension process from Section 3.2 with the active set \mathcal{V}_A consisting of a single vertex $K \in \mathcal{V}$. The four immediate neighbors of K form the set \mathcal{V}_B , and we assign the five control points values on the plane so that \bar{p}_K^K is at the origin, and the points from \mathcal{V}_B are positioned at $(1, 0)$, $(0, 1)$, $(-1, 0)$, and $(0, -1)$. The extension procedure then defines all the other positions required to evaluate $S[\bar{p}^K]$ on the patches near K . In our experience, this results in non-degenerate assignments of \bar{p}^K .

4 Inverse Compositional Method for Subdivision Maps

4.1 Parametric Adjustment

We can now find the optimal parameters q of the parametric adjustment, by minimizing the fit functional with respect to q .

$$J(q) := \int_{\Xi} |I(S[p](\xi)) - T(W(\xi; q))|^2 d\xi$$

We introduce the pointwise error of the current fit as $E(\xi; p) := I(S[p](\xi)) - T(\xi)$, and obtain the following approximation to $J(q)$:

$$J(q) \approx \sum_{m=1}^3 \int_{\Xi} \left[E_m(\xi; p) - \frac{\partial T_m}{\partial \xi^i}(\xi) \gamma_{i'}^{ki}(\xi) q_{i'}^{k'} \right]^2 d\xi,$$

where the subscript $m = 1, 2, 3$ denotes the appropriate color channel.

Differentiating this expression with respect to $q_{i'}^{k'}$ and introducing the notation $h_m^{ki'}(\xi) := \gamma_{i'}^{ki}(\xi) \partial T_m / \partial \xi^i(\xi)$ we get the following system of linear equations for the optimal parametric adjustment parameters q :

$$\int_{\Xi} h_m^{ki'}(\xi) h_m^{lj'}(\xi) d\xi q_l^{j'} = \int_{\Xi} h_m^{ki'}(\xi) E_m(\xi; p) d\xi, \quad k \in \mathcal{V}, i' = 1, 2.$$

As is expected from an inverse compositional method, the matrix $A_{ki',lj'} = \int_{\Xi} h_m^{ki'}(\xi) h_m^{lj'}(\xi) d\xi$ on the left hand side does not depend on p and its inverse can be precomputed, while the right hand side $b_{ki'} = \int_{\Xi} h_m^{ki'}(\xi) E_m(\xi; p) d\xi$ depends on the current error of the fit, and has to be recomputed during optimization.

The linear system $Aq = b$ has $2|\mathcal{V}|$ unknowns, and in order to guarantee its proper solution we need to ensure that the surface template has enough edge features of different orientations (similar to [12]). In order to ensure more robust tracking we apply the multiscale template: in this case we need to ensure that the edge features exist at all the resolutions. Surfaces marked with quad patterns do provide such features as long as the surface patches controlled by a single vertex covers a few of the pattern quads (see Figure 1 for the comparison of condition numbers of the matrix $A = H^t H$).

4.2 Evaluation of Control Vertex Adjustments

Once the parametric adjustment is found, we compute the surface sample displacements that correspond to the approximate inverse of the parametric displacement, namely a surface sample $S[p](\xi)$ is moved to $S[p](W(\xi; -q))$ where q is the optimal parametric displacement parameters found in the previous section.

We find an approximation to the actual samples movement via

$$S[p](W(\xi; -q)) \approx S[p](\xi - q_k^{i'} \gamma_{i'}^{ki}(\xi)) \approx S[p](\xi) - \frac{\partial S[p]}{\partial \xi^i}(\xi) \gamma_{i'}^{ki}(\xi) q_k^{i'},$$

so that the sample with parameter ξ undergoes the displacement

$$\sigma(\xi; q) := -\frac{\partial S[p]}{\partial \xi^i}(\xi) \gamma_{i'}^{ki}(\xi) q_k^{i'};$$

note that $\partial S[p]/\partial \xi$ depends on the current p which makes the update step non-linear [1].

In order to find the approximation to the appropriate displacement of control vertices of the subdivision surface we sample the surface displacement on a fixed four-by-four grid of parameter samples within each quad patch, and solve for the corresponding control vertex displacements that are optimal in the least square sense.

Denote the discrete set of all the sampled surface displacement parameters as Ξ_S . We need to find Δp such that the following set of constraint is approximately satisfied, that is $S[\Delta p](\xi) = \sigma(\xi; q)$ for all $\xi \in \Xi_S$. Using the expression for the subdivision map we obtain $\Delta p_k \phi^k(\xi) = \sigma(\xi; q)$, $\xi \in \Xi_S$. Introduce the matrix of basis function sample values $(\Phi_S)_{\xi v} := \phi^v(\xi)$, $v \in \mathcal{V}_I$, $\xi \in \Xi_S$. The least-squares solution the above linear system then gives us the following expression for Δp :

$$\Delta p = (\Phi_S^t \Phi_S)^{-1} \Phi_S^t \sigma(q),$$

and the matrix inverse on the right-hand side can be precomputed during system initialization. Once the optimal control point adjustment is found, we update the control point positions using $p^{(n+1)} = p^{(n)} + \Delta p$ which concludes a single iteration of our template alignment procedure.

Appearance variation. A simple constant appearance variation model can be added as in [4] by changing the pointwise fit error to be $E(\xi; p) := (I(S[p](\xi)) - I_{average}) - T^*(\xi)$, where $I_{average}$ is an estimate of the average color value of image samples of the current surface, and $T^*(\xi) := T(\xi) - T_{average}$ is the original template adjusted so that its average is zero.

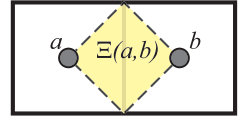
5 Partial Template Matching

When some part of the surface is occluded, it is no longer possible to track its motion, and the template matching should exclude the corresponding control

vertices. At the same time, it would not be practical to create a completely separate surface template from scratch. Rather we would like to reuse a partially active template. In this section, we describe how the template matching algorithm described above can be modified when only a subset of control vertices are allowed to change independently.

We assume that a subset \mathcal{V}_A of active vertices is given (one can think that the surface portion $S(\bigcup_{k \in \mathcal{V}_A} \Xi_k)$ is fully visible); and take $\mathcal{V}_B = \text{bou}(\mathcal{V}_A)$. The union of \mathcal{V}_A and \mathcal{V}_B is therefore used as the set of independent control vertices found by template matching, we call it $\mathcal{V}_{AB} := \mathcal{V}_A \cup \mathcal{V}_B$. Thus, the surface is determined by vertices in \mathcal{V}_{AB} ; we modify the set of warp parameters Δq in the same way, and all the linear systems solved in the algorithm will be of such reduced dimensions.

Another modification to the algorithm is the modification of the integration region Ξ . Denote by \mathcal{E}_{AB} the set of all the edges with both ends in \mathcal{V}_{AB} . With each edge (a, b) we associate the union $\Xi(a, b)$ of two triangular sections of the corresponding patches Ξ_a and Ξ_b , as shown in the figure to the right. The integration region Ξ_{AB} used in the partial matching algorithm is defined to be $\Xi_{AB} = \bigcup_{(a,b) \in \mathcal{E}_{AB}} \Xi(a, b)$. Therefore, the surface map needs to be evaluated on samples within Ξ_{AB} at every step of matching. This is only possible when a wider set of control vertex positions is known; the extension procedure from Section 3.2 is used for extrapolating vertex positions from the set \mathcal{V}_{AB} to the wider set \mathcal{V}_{ext} required for the surface evaluation on Ξ_{AB} .



6 Tracking Quad Patterns with the Multiscale Matching

The derivation of the template alignment method of the previous section relied on the fact that we could take derivatives of the template function and apply basic first order approximations. We would like to apply this method to tracking colored quad patterns similar to the ones used in [13]. Those patterns considered as functions are not even continuous. In this section, we discuss the issues that arise from this complication and our approach to overcoming them. We start by analyzing a simple one-dimensional example, and then discuss the implications of this analysis for the original surface tracking case.

One-dimensional example. Assume that our template is the step function²: $T(x) = \chi(x)$, and consider the error functional $J(p) := \int |T(x+p) - I(x)|^2 dx$. If the image function is a shifted step function, that is $I(x) = \chi(x+a)$ for some a , the error functional is not smooth: $J(p) = |p - a|$. It follows that the gradient based methods may not perform well on such an optimization problem. This can be noticed when we apply template matching on sharp images with discontinuities: the adjustment to the template parameters p coming from the gradient descent method will stop decreasing as it approaches the optimal value.

² Define $\chi(x) = 0$ for $x < 0$ and $\chi(x) = 1$ for $x \geq 0$

In practice we work with a discretized version of the template, so all the derivatives can be evaluated as divided differences. Consider the template at the grid step h :

$$T_h(x) = \begin{cases} 0, & x < -h/2 \\ 1/2 + x/h, & -h/2 \leq x < h/2 \\ 1, & x \geq h/2 \end{cases}$$

The minimization of the functional $J_h(p) := \int |T_h(x+p) - I(x)|^2 dx$ leads to the following expression for the optimal translation of the template:

$$p_h^* = \int (I - T_h)T_h' dx / \int (T_h')^2 dx = \int_{-h/2}^{h/2} (I - T_h) dx.$$

Hence the upper bound on the parameter adjustment is proportional to the step h : $|p_h^*| \leq h\|I - T_h\|_\infty \leq 2h \max\{\|I\|_\infty, 1\}$. Thus, for large adjustments in p we need to employ coarse versions of the template, while small h are preferable for the precise positioning of the template. It therefore makes sense to proceed from coarse to fine discretizations. This is similar in spirit to multiscale optical flow and template matching algorithms [6] [7][14].

We apply the coarse-to-fine approach to our surface template matching algorithm. To illustrate its convergence properties at different resolutions we plot the mean-square error of a template fit with respect to the number of iterations for the *Quad sheet* model (see Figure 1). It is clear that the coarse template matching makes larger adjustments towards the minimum but the precision of the result is limited. At the same time a finer template matching is able to recover the minimum with high precision but requires more iterations, and is also more intensive computationally. The combined method shown in the plot

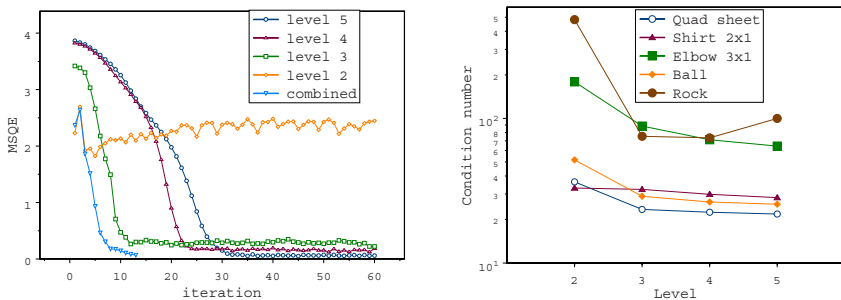


Fig. 1. Left: mean-square error during iterations of template matching procedures at different levels of template resolution. Right: comparison of condition numbers of $H^t H$ matrices at different levels of resolution. For quad patterns the condition number stays relatively low on all the levels, while for natural patterns it is less controlled.

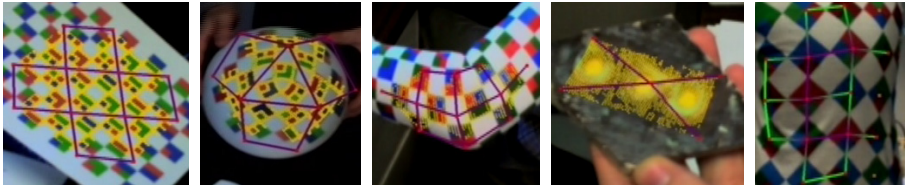


Fig. 2. Various surface templates used. See accompanying video for tracking examples

uses three iterations on each level starting from the coarsest, and achieves better convergence at lower cost. The moderate condition number of the matrix $H^t H$ plotted in Figure 1 also contributes to the success of the multiscale method for quad-marked patterns.

In the case of a piecewise constant quad-marked pattern, the only template samples participating in template matching are located along the discontinuities. Thus, when the template subdivision level l is increased, the number of sampled points will only increase linearly with n rather than quadratically (as n^2 where $n = 2^l$) as is the case for a template with globally non-trivial gradient. This compression of the template improves the efficiency of the matching (see the left two templates in Figure 2, the yellow dots indicate the used template samples).

7 Results

We have implemented our template matching algorithm for Doo-Sabin subdivision surfaces within a real-time tracking application framework. After manual initialization, the template is tracked in video sequence. A linear temporal prediction scheme is employed to obtain the initial guess for the template positioning in every consecutive frame of video. The application was able to perform at 30 frames per second for all the full surface tracking examples presented below on a 2GHz Pentium laptop. The video was acquired with a digital camera at 640×480 resolution. We used the combined multiscale method that ran two iterations of matching on each level of resolution on levels two to four, and a single iteration on level five. The below table shows the number of inside and boundary control vertices in the models used for this paper. The accompanying video contains video sequences captured in real time.

Name	Number of CVs	Number of active inside CVs	Number of active boundary CVs	Texture type
Quad sheet	12	4	8	pure
Ball	9	3	6	pure
Shirt 2x1	8	2	6	acquired
Elbow 3x1	11	3	8	acquired
Rock	5	1	4	acquired
Partial shirt	21	5	10	pure

We have used both predefined *pure* quad template patterns and the textures *acquired* from the video frame during the initialization process. The pure patterns result in the sparse pattern for the participating samples along the quad edges; for the acquired textures we used thresholding on the $h_m^{li}(\xi)$ coefficients to determine which samples should be participating in the integral discretization.

For the partial tracking example, we used a five by five grid template for the t-shirt. The active region included five vertices as shown in Figure 2. This example runs at 15 frames per second.

8 Future Work

We presented a multiscale method for matching subdivision surface templates. The future work will need to address the maintenance of the active visible region for partial surface tracking as well as the automatic initialization procedure. The compositional methods work within the surface and cannot account for surface displacement near its silhouettes from a single view. A multi-view extension of the presented procedure can help alleviate this problem. A more complex appearance variation modeling is also left as a future work direction.

Acknowledgments. This work was partially supported by NSF(CCR-0133554) and University of Michigan AI Lab.

References

1. Matthews, I., Baker, S.: Active appearance models revisited. Technical Report CMU-RI-TR-03-02, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (2003)
2. Cascia, M.L., Sclaroff, S., Athitsos, V.: Fast, reliable head tracking under varying illumination: An approach based on robust registration of texture-mapped 3d models. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)* **22** (2000)
3. Zorin, D., Schröder, P., eds.: Subdivision for Modeling and Animation. Course Notes. ACM SIGGRAPH (1999)
4. Baker, S., Matthews, I.: Equivalence and efficiency of image alignment algorithms. In: *Proc. of the CVPR*. (2001)
5. Szeliski, R., Shum, H.Y.: Motion estimation with quadtree splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18** (1996) 1199–1210
6. Simoncelli, E.: Bayesian multi-scale differential optical flow. In: *Handbook of Computer Vision and Applications*. (1993) 128–129
7. Borgefors, G.: Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10** (1988) 849–865
8. Sclaroff, S., Isidoro, J.: Active blobs. In: *Proceedings of ICCV 98*. (1998) 1146–1153
9. Doo, D., Sabin, M.: Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design* **10** (1978) 356–360
10. Warren, J., Weimer, H.: *Subdivision Methods For Geometric Design: A Constructive Approach*. Morgan Kaufmann (2001)

11. Zorin, D., Schröder, P.: A unified framework for primal/dual quadrilateral subdivision schemes. *CAGD* **18** (2001) 429–454
12. Shi, J., Tomasi, C.: Good features to track. In: *CVPR*. (1994) 593–600
13. Guskov, I., Klibanov, S., Bryant, B.: Trackable surfaces. In: *Proceedings of ACM/EG Symposium on Computer Animation*. (2003) 251–257
14. Gleicher, M.: Projective registration with difference decomposition. In: *Proc. of IEEE CVPR 1997*. (1997) 331–337

A Fourier Theory for Cast Shadows

Ravi Ramamoorthi¹, Melissa Koudelka², and Peter Belhumeur¹

¹ Columbia University, {ravir, belhumeur}@cs.columbia.edu

² Yale University, melissa.koudelka@yale.edu

Abstract. Cast shadows can be significant in many computer vision applications such as lighting-insensitive recognition and surface reconstruction. However, most algorithms neglect them, primarily because they involve non-local interactions in non-convex regions, making formal analysis difficult. While general cast shadowing situations can be arbitrarily complex, many real instances map closely to canonical configurations like a wall, a V-groove type structure, or a pitted surface. In particular, we experiment on 3D textures like moss, gravel and a kitchen sponge, whose surfaces include canonical cast shadowing situations like V-grooves. This paper shows theoretically that many shadowing configurations can be mathematically analyzed using convolutions and Fourier basis functions. Our analysis exposes the mathematical convolution structure of cast shadows, and shows strong connections to recently developed signal-processing frameworks for reflection and illumination. An analytic convolution formula is derived for a 2D V-groove, which is shown to correspond closely to many common shadowing situations, especially in 3D textures. Numerical simulation is used to extend these results to general 3D textures. These results also provide evidence that a common set of illumination basis functions may be appropriate for representing lighting variability due to cast shadows in many 3D textures. We derive a new analytic basis suited for 3D textures to represent illumination on the hemisphere, with some advantages over commonly used Zernike polynomials and spherical harmonics. New experiments on analyzing the variability in appearance of real 3D textures with illumination motivate and validate our theoretical analysis. Empirical results show that illumination eigenfunctions often correspond closely to Fourier bases, while the eigenvalues drop off significantly slower than those for irradiance on a Lambertian curved surface. These new empirical results are explained in this paper, based on our theory.

1 Introduction

Cast shadows are an important feature of appearance. For instance, buildings may cause the sun to cast shadows on the ground, the nose can cast a shadow onto the face, and local concavities in rough surfaces or textures can lead to interesting shadowing effects. However, most current vision algorithms do not explicitly consider cast shadows. The primary reason is the difficulty in formally analyzing them, since cast shadows involve non-local interactions in concave regions.

In general, shadowing can be very complicated, such as sunlight passing through the leaves of a tree, and mathematical analysis seems hopeless. However, we believe many common shadowing situations have simpler structures, some of which are illustrated

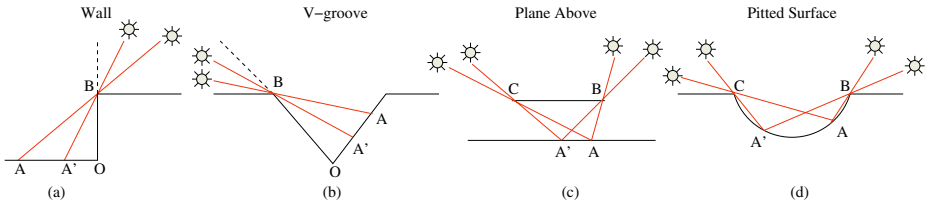


Fig. 1. Four common shadowing situations. We show that these all have similar structures, amenable to treatment using convolution and Fourier analysis. The red lines indicate extremal rays, corresponding to shadow boundaries for distant light sources.

in Figure 1. From left to right, shadowing by a wall, a V-groove like structure, a plane such as a desk above, and a pitted or curved surface. Though the figure is in 2D, similar patterns often apply in 3D along the radial direction, with little change in the extent of shadowing along transverse or azimuthal directions.

Our theory is motivated by some surprising practical results. In particular, we focus on the appearance of natural 3D textures like moss, gravel and kitchen sponge, shown in Figures 2 and 6. These objects have fine-scale structures similar to the canonical configurations shown in Figure 1. Hence, they exhibit interesting illumination and view-dependence, which is often described using a bi-directional texture function (BTF) [3]. In this paper, we analyze lighting variability, assuming fixed view. Since these surfaces are nearly flat and diffuse, one might expect illumination variation to correspond to simple Lambertian cosine-dependence. However, cast shadows play a major role, leading to effects that are quantitatively described and mathematically explained here.

We show that in many canonical cases, cast shadows have a simple convolution structure, amenable to Fourier analysis. This indicates a strong link between the mathematical properties of visibility, and those of reflection and illumination (but ignoring cast shadows) for which Basri and Jacobs [1], and Ramamoorthi and Hanrahan [15, 16], have recently derived signal-processing frameworks. In particular, they [1, 15] show that the irradiance is a *convolution* of the lighting and the *clamped cosine* Lambertian reflection function. We derive an analogous result for cast shadows, as convolution of the lighting with a *Heaviside step* function. Our results also generalize Soler and Sillion's [19] convolution result for shadows when source, blocker and receiver are all in parallel planes—for instance, V-grooves (b in Figure 1, as well as a and d) do not contain any parallel planes. Our specific technical contributions include the following:

- We derive an analytic convolution formula for a 2D V-groove, and show that it applies to many canonical shadowing situations, such as those in Figure 1.
- We analyze the illumination eigenmodes, showing how they correspond closely to Fourier basis functions. We also analyze the eigenvalue spectrum, discussing similarities and differences with convolution results for Lambertian curved surfaces and irradiance, and showing why the falloff is slower in the case of cast shadows.
- We explain important lighting effects in 3D textures, documented quantitatively here for the first time. Experimental results confirm the theoretical analysis.
- We introduce new illumination basis functions over the hemisphere for lighting variability due to cast shadows in 3D textures, potentially applicable to compression,

interpolation and prediction. These bases are based on analytic results and numerical simulation, and validated by empirical results. They have some advantages over the commonly used spherical harmonics and Zernike polynomials.

Our paper builds on a rich history of previous work on reflection models, such as Oren-Nayar [12], Torrance-Sparrow [21], Wolff et al. [23] and Koenderink et al. [6], as well as several recent articles on the properties of 3D textures [2,20]. Our analytic formulae are derived considering the standard V-grooves used in many of these previous reflection models [12,21]. Note that many of these models include a complete analysis of visibility in V-grooves or similar structures, for any single light source direction. We differ in considering cast shadows because of complex illumination, deriving a convolution framework, and analyzing the *eigenstructure* of visibility. Our work also relates to recent approaches to real-time rendering, such as the precomputed transfer method of Sloan et al. [18], that represents appearance effects including cast shadows, due to low-frequency illumination, represented in spherical harmonics. However, there is no analytic convolution formula or insight in their work as to the optimal basis functions or the number of terms needed for good approximation. We seek to put future real-time rendering methods on a strong theoretical footing by formalizing the idea of convolution for cast shadows, analyzing the form of the eigenvalue spectrum, showing that the decay is much slower than for Lambertian irradiance, and that we therefore need many more basis functions to capture sharp shadows than the low order spherical harmonics and polynomials used by Sloan et al. [18] and Malzbender et al. [11].

2 The Structure of Cast Shadows

In this section, we briefly discuss the structure of cast shadows, followed in the next section by a derivation of an analytic convolution formula for a 2D V-groove, Fourier and principal component analysis, and initial experimental observations and validation.

First, we briefly make some theoretical observations. Consider Figures 1 a and b. There is a *single extreme point* B . As we move from O to A' to A (with the extremal rays being OB , $A'B$ and AB), the visible region of the illumination ***monotonically increases***. This local shadowing situation, with a single extreme point B , and monotonic variation of the visible region of the illumination as one moves along the surface, is one of the main ideas in our derivation. Furthermore, multiple extreme points or blockers can often be handled independently. For instance, in Figures 1 c and d, we have two extreme points B and C . The net shadowing effect is essentially just the superposition of the effects of extreme rays through B and C .

Second, we describe some new experimental results on the variability of appearance in 3D textures with illumination, a major component of which are cast shadowing interactions similar to the canonical examples in Figure 1. In Figure 2, we show an initial experiment. We illuminated a sample of gravel along an arc (angle ranged from -90° to $+64^\circ$, limited by specifics of the acquisition). The varying appearance with illumination clearly suggests cast shadows are an important visual feature. The figure also shows a conceptual diagrammatic representation of the profile of a cross-section of the surface, with many points shadowed in a manner similar to Figure 1 (a), (b) and (d).

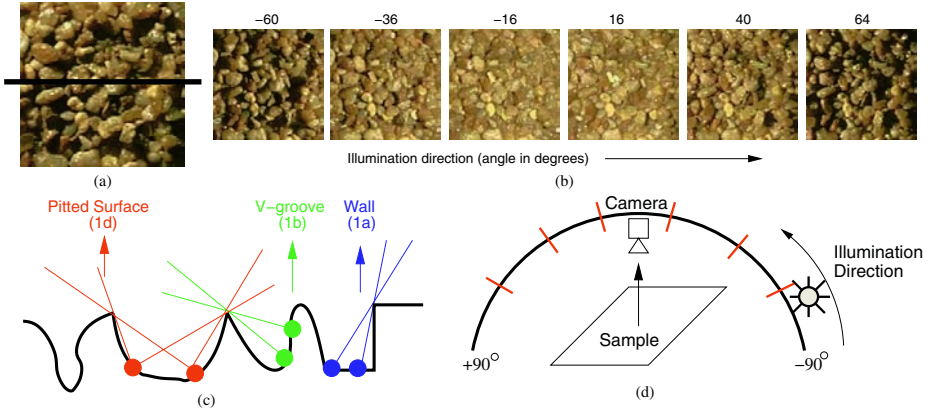


Fig. 2. (a): Gravel texture, which exhibits strong shadowing. (b): Images with different light directions clearly show cast shadow appearance effects, especially at large angles. The light directions correspond to the red marks in (d). (c): Conceptual representation of a profile of a cross section through surface (drawn in black in a). (d): Schematic of experimental setup.

3 2D Analysis of Cast Shadows

For mathematical analysis, we begin in flatland, i.e., a 2D slice through the viewpoint. We will consider a V-groove model, shown in Figure 3, corresponding to Figure 1 b. However, the derivation will be similar for any other shadowing situation, such as those in Figure 1, where the visibility is locally *monotonically changing*. Note that the V-groove model in Figure 3 can model the examples in Figures 1 a and b ($\beta_1 = 0, \beta_2 = \pi/2$ and $\beta_1 = \beta_2$), and each of the extreme points of Figures 1 c and d.

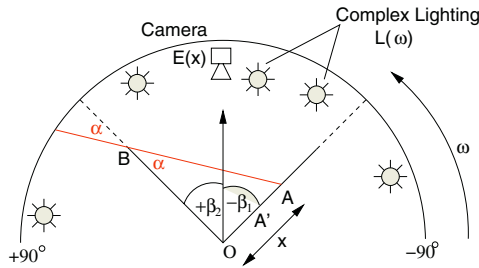


Fig. 3. Diagram of V-groove with groove angle ranging from $-\beta_1$ to $+\beta_2$. While the figure shows $\beta_1 = \beta_2$, as is common for previous V-groove models, there is no requirement of symmetry. We will be interested in visibility for points $A(x)$ where x is the distance along the groove (the labels are as in Figure 1; the line $A'B$ is omitted for clarity). Note that the visible region of $A(x)$, determined by $\alpha(x)$, increases monotonically with x along the groove.

3.1 Convolution Formula for Shadows in a V-Groove

Our goal is to find the irradiance¹ $E(x, \beta)$ as a function of groove angle $\beta = [-\beta_1, +\beta_2]$, and the distance along the groove x . Without loss of generality, we consider the right side of the groove only. The left side can be treated similarly. For a particular groove (fixed β), pixels in a single image correspond directly to different values of x , and the irradiance $E(x)$ is directly proportional to pixel brightness.

$$E(x, \beta) = \int_{-\pi/2}^{\pi/2} L(\omega) V(x, \omega, \beta) d\omega, \quad (1)$$

where $L(\omega)$ is the incident illumination intensity, which is a function of the incident direction ω . We make no restrictions on the lighting, except that it is assumed distant, so the angle ω does not depend on location x . This is a standard assumption in environment map rendering in graphics, and has been used in previous derivations of analytic convolution formulae [1,16]. V is the binary visibility in direction ω at location $A(x)$.

Monotonic Variation of Visibility: As per the geometry in Figure 3, the visibility is 1 in the range from $-\beta_1$ to $\beta_2 + \alpha(x)$ and 0 or (cast) shadowed otherwise. It is important to note that $\alpha(x)$ is a *monotonically increasing* function of x , i.e., the portion of the illumination visible increases as one moves along the right side of the groove from O to A' to A (with corresponding extremal rays OB , $A'B$ and AB).

Reparameterization by α : We now simply use α to parameterize the V-groove. This is just a change of variables, and is valid as long as α *monotonically varies* with x . Locally, α is always proportional to x , since we may do a local Taylor series expansion, keeping only the first or linear term.

Representation of Visibility: We may now write down the function $V(x, \omega, \beta)$ newly reparameterized as $V(\alpha, \omega, \beta)$. Noting that V is 1 only in the range from $[-\beta_1, \beta_2 + \alpha]$,

$$\begin{aligned} V(\alpha, \omega, \beta) &= H(-\beta_1 - \omega) - H((\beta_2 + \alpha) - \omega), \\ H(u) &= 1 \text{ if } u < 0, \quad 0 \text{ if } u > 0, \end{aligned} \quad (2)$$

where $H(u)$ is the Heaviside step function. The first term on the right hand side zeros the visibility when $\omega < -\beta_1$ and the second term when $\omega > \beta_2 + \alpha$. Figure 4 illustrates this diagrammatically. In the limit of a perfectly flat Lambertian surface, $\beta_1 = \beta_2 = \pi/2$, and $\alpha = 0$. In that case, the first term on the right of Equation 2 is always 1, the second term is 0, and $V = 1$ (no cast shadowing), as expected.

For a particular groove (fixed β), V is given by the following intervals.

$$\begin{array}{lll} -\pi/2 < \omega < -\beta_1 & V = 0 & \text{independent of } \alpha \\ -\beta_1 < \omega < +\beta_2 & V = 1 & \text{independent of } \alpha \\ +\beta_2 < \omega < \beta_2 + \alpha & V = 1 & \text{interval depends on } \alpha \\ \beta_2 + \alpha < \omega < \pi/2 & V = 0 & \text{interval depends on } \alpha. \end{array} \quad (3)$$

¹ Since we focus on cast shadows, we will assume Lambertian surfaces, and will neglect the incident cosine term. This cosine term may be folded into the illumination function if desired, as the surface normal over a particular face (side) of the V-groove is constant.

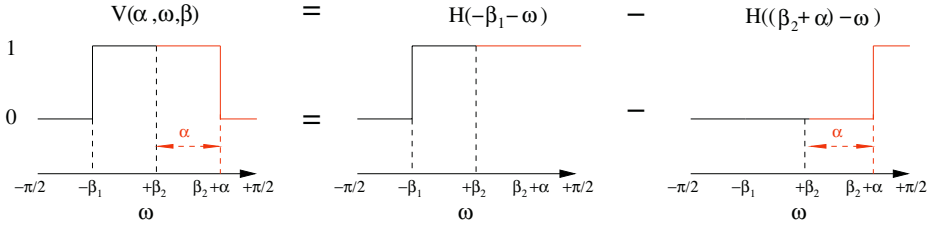


Fig. 4. Illustration of the visibility function as per Equation 2. The black portions of the graphs where $\omega < +\beta_2$ are independent of α or groove location, while the red portions with $\alpha > +\beta_2$ vary linearly with α , leading to the convolution structure.

Convolution Formula: Plugging Equation 2 back into Equation 1, we obtain

$$E(\alpha, \beta) = \int_{-\pi/2}^{\pi/2} L(\omega) H(-\beta_1 - \omega) d\omega - \int_{-\pi/2}^{\pi/2} L(\omega) H((\beta_2 + \alpha) - \omega) d\omega. \quad (4)$$

E is the sum of two terms, the first of which depends only on groove angle β_1 , and the second that also depends on groove location or image position α . In the limit of a flat diffuse surface, the second term vanishes, while the first corresponds to convolution with unity, and is simply the (unshadowed) irradiance or integral of the illumination. We now separate the two terms to simplify this result as (\otimes is the convolution operator)

$$E(\alpha, \beta) = \tilde{E}(-\beta_1) - \tilde{E}(\beta_2 + \alpha) \\ \tilde{E}(u) = \int_{-\pi/2}^{\pi/2} L(\omega) H(u - \omega) d\omega = L \otimes H. \quad (5)$$

Fourier Analysis: Equation 5 makes clear that the net visibility or irradiance is a simple *convolution* of the incident illumination with the Heaviside step function that accounts for cast shadow effects. This is our main analytic result, deriving a new convolution formula that sheds theoretical insight on the structure of cast shadows. It is therefore natural to also derive a product formula in the Fourier or frequency domain,

$$\tilde{E}_k = \sqrt{\pi} L_k H_k, \quad (6)$$

where L_k are the Fourier illumination coefficients, and H_k are Fourier coefficients of the Heaviside step function, plotted in Figure 5. The even coefficients H_{2k} vanish, while the odd **coefficients decay** as $1/k$. The analytic formula is

$$k = 0 : H_0 = \frac{\sqrt{\pi}}{2} \\ \text{odd } k : H_k = \frac{i}{\sqrt{\pi} k}. \quad (7)$$

3.2 Eigenvalue Spectrum and Illumination Eigenmodes for Cast Shadows

Our convolution formula is conceptually quite similar to the convolution formula and signal-processing analysis done for convex curved Lambertian surfaces or irradiance

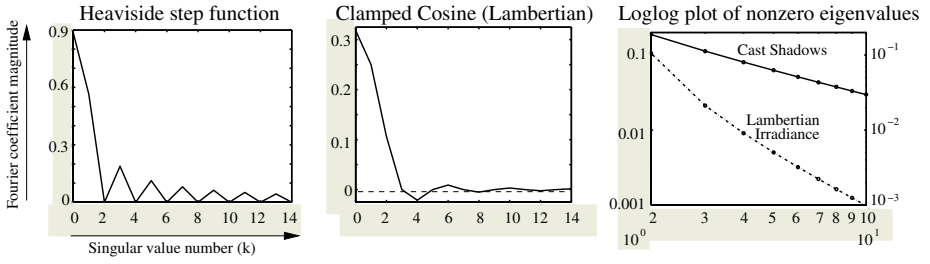


Fig. 5. Comparison of Fourier coefficients for the Heaviside step function for cast shadows (left) and the clamped cosine Lambertian function for irradiance (middle). For the step function, even terms vanish, while odd terms decay as $1/k$. For the clamped cosine, odd terms greater than 1 vanish, while even terms decay much faster as $1/k^2$. On the right is a loglog plot of the absolute values of the nonzero eigenvalues. The graphs are straight lines with slope -1 for cast shadows, compared to the quadratic decay (slope -2) for irradiance.

by Basri and Jacobs [1] and Ramamoorthi and Hanrahan [14,15]. In this subsection, we analyze our results further in terms of the illumination eigenmodes that indicate the lighting distributions that have the most effect, and the corresponding eigenvalues or singular values that determine the relative importance of the modes. We also compare to similar analyses for irradiance on a curved surface [1,13,14,15].

Illumination eigenmodes are usually found empirically by considering the SVD of a large number of images under different (directional source) illuminations, as in lighting-insensitive face and object recognition [4,5]. It seems intuitive in our case that the eigenfunctions will be sines and cosines. To formalize this analytically, we must relate the convolution formula above, that applies to a single image with complex illumination, to the eigenfunctions derived from a number of images taken assuming directional source lighting. Our approach is conceptually similar to Ramamoorthi’s work on analytic PCA construction [13] for images of a convex curved Lambertian object.

Specifically, we analyze $V(\alpha, \omega, \beta)$ for a particular groove (fixed β). Then, $V(\alpha, \omega)$ is a matrix with rows corresponding to groove locations (image pixels) α and columns corresponding to illumination directions ω . A singular-value decomposition (SVD) will give the eigenvalues (singular values) and illumination eigenmodes. It can be formally shown (details omitted here) that the following results hold, as expected.

Eigenvalue Spectrum: The eigenvalues decay as $1/k$, corresponding to the Heaviside coefficients, as shown in Figure 5. Because of the relatively slow $1/k$ decay, we need quite high frequencies (many terms) for good approximation of cast shadows. On the other hand², for irradiance on a convex curved surface, we convolve with the clamped cosine function $\max(\cos \theta, 0)$ whose Fourier coefficients falloff quadratically as $1/k^2$, with very few terms needed for accurate representation [1,15].

In actual experiments on 3D textures, the eigenvalues decay somewhat faster. First, as explained in section 4.1, the eigenvalues for cast shadows decay as $1/k^{3/2}$ (loglog

² The Heaviside function has a position or C^0 discontinuity at the step, while the clamped cosine has a derivative or C^1 discontinuity at $\cos \theta = 0$. It is known in Fourier analysis [10] that a C^n discontinuity will generally result in a spectrum that falls off as $1/k^{n+1}$.

slope -1.5) in 3D, as opposed to $1/k$ in 2D. Second, in the Lambertian case, since we are dealing with flat, as opposed to spherical surfaces, the eigenvalues for irradiance drop off much faster than $1/k^2$. In fact, for an ideal flat diffuse surface, all of the energy is in the first eigenmode, that corresponds simply to Lambertian cosine-dependence.

Illumination Eigenmodes: The illumination eigenmodes are simply Fourier basis functions—sines and cosines. This is the case for irradiance on a curved surface in 2D as well [14], reinforcing the mathematically similar convolution structure.

Implications: There are many potential implications of these results, to explain empirical observations and devise practical algorithms. For instance, it has been shown [15] that illumination estimation from a convex Lambertian surface is ill-posed since only the first two orders can be estimated. But Sato et al. [17] have shown that illumination can often be estimated from cast shadows. Our results explain why it is feasible to estimate much higher frequencies of the illumination from the effects of cast shadows. In lighting-insensitive recognition, there has been much work on low-dimensional subspaces for Lambertian objects [1,4,5,13]. Similar techniques might be applied, simply using more basis functions, and including cast shadow effects, since cast shadows and irradiance have the same mathematical structure. Our results have direct implications in BTF modeling and rendering for representing illumination variability, and providing appropriate basis functions for compression and synthesis.

3.3 Experimental Validation

In this subsection, we present an initial quantitative experimental result motivating and validating our derivation. The next sections generalize these results to 3D, and present more thorough experimental validations. We used the experimental setup of Figure 2, determining the eigenvalue spectrum and illumination eigenmodes for both a sample of moss, and a flat piece of paper. The paper serves as a control experiment on a nearly Lambertian surface. Our results are shown in Figure 6.

Eigenvalue Spectrum: As seen in Figure 6 (c), the eigenvalues (singular values) for moss when plotted on a log-log scale lie on a straight line with slope approximately -1.5, as expected. This contrasts with the expected result for a flat Lambertian surface, where we should in theory see a single eigenmode (simply the cosine term). Indeed, in our control experiment with a piece of paper, also shown in Figure 6 (c), 99.9% of the energy for the paper is in the first eigenmode, with a very fast decay after that.

Illumination Eigenmodes: As predicted, the illumination eigenmodes are simply Fourier basis functions—sines and cosines. This indicates that *a common set of illumination eigenfunctions* may describe lighting-dependence in many 3D textures.

4 3D Numerical Analysis of Cast Shadows

In 3D, V-grooves can be rotated to any orientation about the vertical; hence, the *direction* of the Fourier basis functions can also be rotated. For a *given V-groove direction*, the

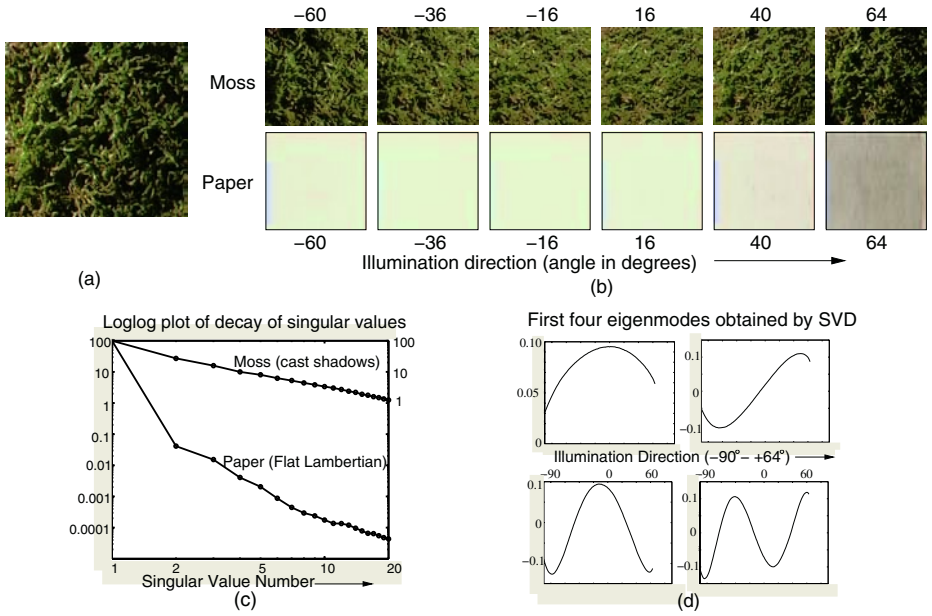


Fig. 6. (a): Moss 3D texture with significant shadowing. Experimental setup is as in Figure 2. (b): 6 images of the moss with different lighting directions, as well as a control experiment of paper (a flat near-Lambertian surface). Note the variation of appearance of moss with illumination direction due to cast shadows, especially for large angles. In contrast, while the overall intensity changes for the paper, there is almost no variation on the surface. (c): Decay of singular values for illumination eigenmodes for 3D textures is a straight line with slope approximately -1.5 on a logarithmic scale. In contrast, for a flat near-Lambertian surface, all of the energy is in the first eigenmode with a very rapid falloff. (d): The first four illumination eigenfunctions for moss, which are simply sines and cosines.

2D derivation essentially still holds, since it depends on the monotonic increase in visibility as one moves along the groove, which still holds in 3D. The interesting question is, what is the set of illumination basis functions that encompasses all V-groove (and correspondingly Fourier) orientations in 3D?

One might expect the basis functions to be close to spherical harmonics [9], the natural extension of the Fourier basis to the sphere. However, we are considering only the visible upper hemisphere, and we will see that our basis functions take a somewhat simpler form than spherical harmonics or Zernike polynomials [7], corresponding closely to 2D Fourier transforms. In this section, we report on the results of numerical simulations, shown in Figures 7, 8 and 9. We then verify these results with experiments on real 3D textures including moss, gravel and a kitchen sponge.

4.1 Numerical Eigenvalue Spectrum and Illumination Eigenmodes

For numerical simulation, we consider V-grooves oriented at (rotated by) arbitrary angles about the vertical, ranging from 0 to 2π . For each orientation, we consider a number

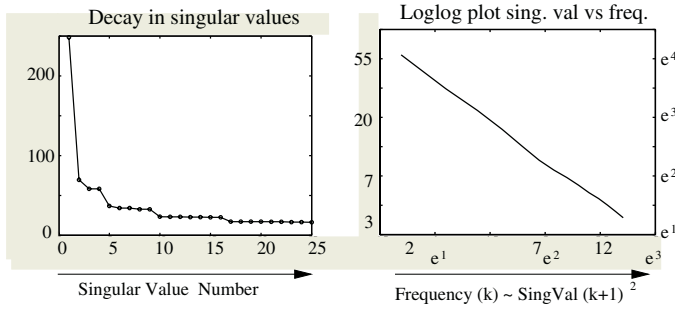


Fig. 7. *Left:* Singular values for illumination basis functions due to cast shadows in a simulated 3D texture (randomly oriented V-grooves), plotted on a linear scale. A number of singular values cluster together. *Right:* Decay of singular values [value vs *frequency* or *square root* of singular value number] on a logarithmic scale (with natural logarithms included as axis labels). We get a straight line with slope approximately -1.5 as expected.

of V-groove angles with β ranging from 0 to $\pi/2$. In essence, we have an ensemble of a large number of V-grooves (1000 in our simulations). Each point on each V-groove has a binary visibility value for each point on the illumination hemisphere. We can assemble all this information into a large visibility matrix, where the rows correspond to V-groove points (image pixels), and the columns to illumination directions. Then, as in experiments with real textures like Figure 6, we do an SVD³ to find the illumination eigenmodes and eigenvalues.

Numerical Eigenvalue Spectrum: We first consider the eigenvalues or singular values, plotted in the left of Figure 7 on a linear scale. At first glance, this plot is rather surprising. Even though the singular values decrease with increasing frequency, a number of them cluster together. Actually, these results are very similar to those for irradiance and spherical harmonics [1,13,15], where $2k + 1$ basis functions of order k are similar. Similarly, our eigenmodes are Fourier-like, with $2k + 1$ eigenmodes at order k (with a total of $(k + 1)^2$ eigenmodes up to order k). Therefore, to determine the decay of singular values, it is more appropriate to consider them as a function of order k . We show k ranging from 1 to 15 in the right of Figure 7.

As expected, the curve is almost exactly a straight line on a log-log plot, with a slope of approximately -1.5. The higher slope (**-1.5 compared to -1 in 2D**) is a natural consequence of the properties of Fourier series of a function with a curve discontinuity [10], as is the case in 3D visibility. The total energy (sum of *squared* singular values) at each order k goes as $1/k^2$ in both 2D and 3D cases. However, in 3D, each frequency band contains $2k + 1$ functions, so the energy in each individual basis function decays as $1/k^3$, with the singular values therefore falling off as $1/k^{3/2}$.

Numerical Illumination Eigenmodes: The first nine eigenmodes are plotted in Figure 8, where we label the eigenmodes using (m, n) with the net frequency given by

³ Owing to the large size of the matrices both here and in our experiments with real data, SVD is performed in a 2 step procedure in practice. First, we find the basis functions and eigenvalues for each V-groove. A second SVD is then performed on these weighted basis functions.

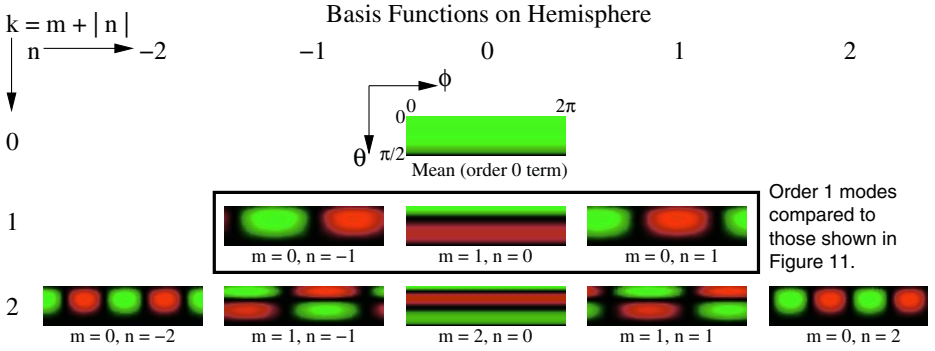


Fig. 8. 3D hemispherical basis functions obtained from numerical simulations of V-grooves. Green denotes positive values and red denotes negative values. θ and ϕ are a standard spherical parameterization, with the cartesian $(x, y, z) = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$.

$k = m + |n|$, with $k \geq 0$, $-k \leq n \leq k$, and $m = k - |n|$. This labeling anticipates the ensuing discussion, and is also quite similar to that used for spherical harmonics.

To gain further insights, we attempt to *factor* these basis functions into a separable form. Most commonly used 2D or (hemi)spherical basis functions are factorizable. For instance, consider the 2D Fourier transform. In this case,

$$W_{mn}(x, y) = U_m(x)V_n(y), \quad (8)$$

where W is the (complex) 2D basis function $\exp(imx)\exp(iny)$, and U_m and V_n are 1D Fourier functions ($\exp(imx)$ and $\exp(iny)$ respectively). Spherical harmonics and Zernike polynomials are also factorizable, but doing so is somewhat more complicated.

$$W_{mn}(\theta, \phi) = U_n^m(\theta)V_n(\phi), \quad (9)$$

where V_n is still a Fourier basis function $\exp(in\phi)$ [this is because of azimuthal symmetry in the problem and will be true in our case too], and U_n^m are associated Legendre polynomials for spherical harmonics, or Zernike polynomials. Note that U_n^m now has two indices, unlike the simpler Fourier case, and also depends on azimuthal index n .

We now factor our eigenmodes. The first few eigenfunctions are almost completely factorizable, and representable in a form similar to Equation 8, i.e., like a 2D Fourier transform, and simpler⁴ than spherical harmonics or Zernike polynomials,

$$W_{mn}(\theta, \phi) = U_m(\theta)V_n(\phi). \quad (10)$$

Figure 9 shows factorization into 1D functions $U_m(\theta)$ and $V_n(\phi)$. It is observed that the U_m correspond closely to odd Legendre polynomials P_{2m+1} . This is not surprising since Legendre polynomials are spherical frequency-space basis functions. We observe

⁴ Mathematically, functions of the form of Equation 10 can have a discontinuity at the pole $\theta = 0$. However, in our numerical simulations and experimental tests, we have found that this form closely approximates observed results, and does not appear to create practical difficulties.

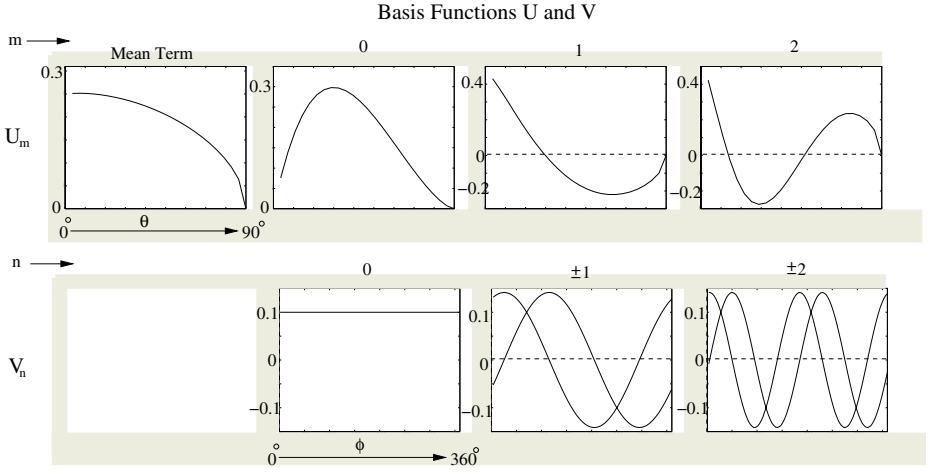


Fig. 9. The functions in Figure 8 are simple products of 1D basis functions along elevation θ and azimuthal ϕ directions, as per Equation 10. Note that the $V_{\pm n}$ are sines and cosines while the U_m are approximately Legendre polynomials (P_3 for $m = 1$, P_5 for $m = 2$). Figure 12 shows corresponding experimental results on an actual 3D texture.

only odd terms $2m + 1$, since they correctly vanish at $\theta = \pi/2$, when a point is always shadowed. V_n are simply Fourier azimuthal functions or sines and cosines. The net frequency $k = m + |n|$, with there being $2k + 1$ basis functions at order k .

4.2 Results of Experiments with Real 3D Textures

In this subsection, we report on empirical results in 3D, showing that the experimental observations are consistent with, and therefore validate, the theoretical and numerical analysis. We considered three different 3D textures—the moss and gravel, shown in Figure 6, and a kitchen sponge, shown in Figure 14. We report in this section primarily on results for the sponge; results for the other samples are similar.

For each texture, we took a number of images with a fixed overhead camera view, and varying illumination direction. The setup in Figure 2 shows a 2D slice of illumination directions. For the experiments in this section, the lighting ranged over the full 3D hemisphere. That is, θ ranged from $[14^\circ, 88^\circ]$ in 2 degree increments (38 different elevation angles) and ϕ from $[-180^\circ, 178^\circ]$ also in 2 degree increments (180 different azimuthal angles). The acquisition setup restricted imaging near the pole. Hence we captured 6840 images (38×180) for each texture. This is a two order of magnitude denser sampling than the 205 images acquired by Dana et al. [3] to represent both light and view variation, and provides a good testbed for comparison with simulations.

For numerical work, we then assembled all of this information in a large matrix, the rows of which were image pixels, and the columns of which were light source directions. Just as in our numerical simulations, we then used SVD to find the illumination eigenmodes and eigenvalues. We validate the numerical simulations by comparing the experimental results for real data to the expected (i.e., numerical) results just described.

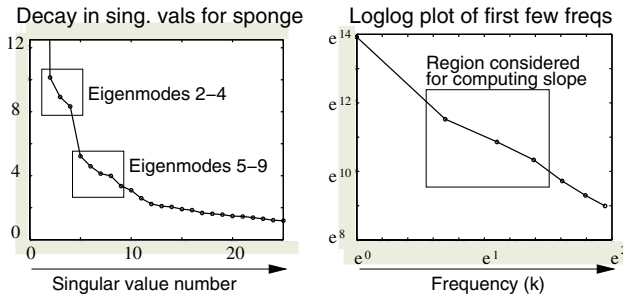


Fig. 10. *Left:* Plot of singular values for the sponge on a linear scale. *Right:* Singular values vs frequencies on a logarithmic scale, with natural log axis labels. These experimental results should be compared to the predicted results from numerical simulation in Figure 7.

Experimental eigenvalue spectrum: Figure 10 plots the experimentally observed falloff of eigenvalues. We see on the left that eigenmodes 2-4 (the first three after the mean term) cluster together, as predicted by our numerical simulations. One can see a rather subtle effect of clustering in second order eigenmodes as well, but beyond that, the degeneracy is broken. This is not surprising for real data, and consistent with similar results for PCA analysis in Lambertian shading [13].

Computing the slope for singular value dropoff is difficult because of insufficiency of accurate data (the first 20 or so eigenmodes correspond only to the first 5 orders, and noise is substantial for higher order eigenmodes). For low orders (corresponding to eigenmodes 2-16, or orders 1-3), the slope on a loglog plot is approximately -1.6, as shown in the right of Figure 10, in agreement with the expected result of -1.5.

Experimental illumination eigenmodes: We next analyze the forms of the eigenmodes; the order 1 modes for moss, gravel and sponge are shown in Figure 11. The first order eigenmodes observed are linear combinations of the actual separable functions—this is expected, and just corresponds to a rotation.

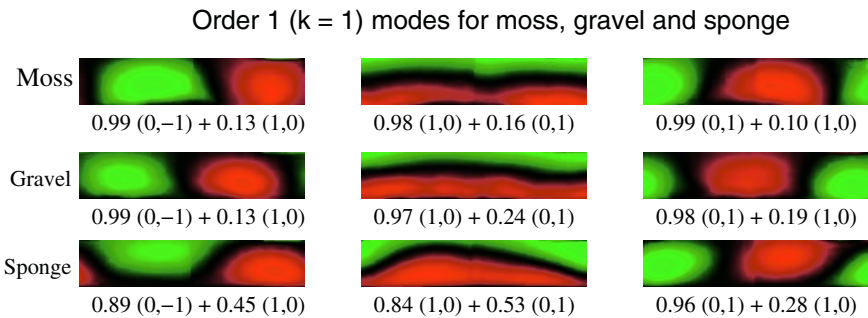


Fig. 11. Order 1 eigenmodes experimentally observed for moss, gravel and sponge. Note the similarity between the 3 textures, and to the basis functions in Figure 8. The numbers below represent each eigenmode as a linear combination of separable basis functions.

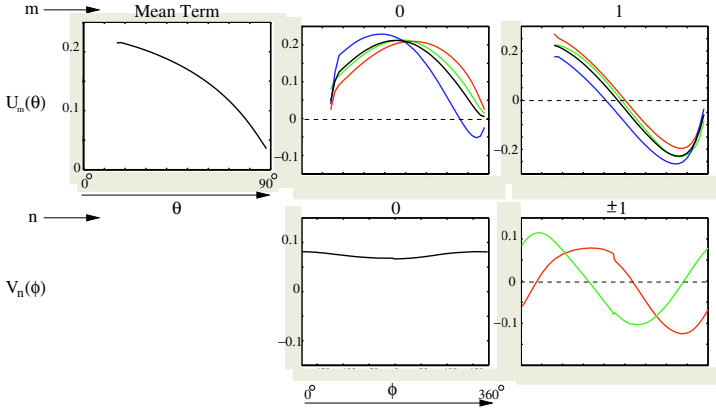


Fig. 12. Factored basis functions $U_m(\theta)$ and $V_n(\phi)$ for sponge. The top row shows the mean eigenmode, and the functions $U_0(\theta)$ and $U_1(\theta)$. Below that are the nearly constant $V_0(\phi)$ and the sinusoidal $V_1(\phi)$, $V_{-1}(\phi)$. The colors red, blue and green respectively are used to refer to the three order 1 eigenmodes that are factored to obtain U_m and V_n . We use black to denote the mean value across the three eigenmodes. It is seen that all the eigenmodes have very similar curves, which also match the results in Figure 9.

We next found the separable functions U_m and V_n along θ and ϕ by using an SVD of the 2D eigenmodes. As expected, the U and V basis functions found separately from the three order 1 eigenmodes were largely similar, and matched those obtained from numerical simulation. Our plots in Figure 12 show both the *average* basis functions (in black), and the individual functions from the three eigenmodes (in red, blue and green) for the sponge dataset. We see that these have the expected forms, and the eigenmodes are well described as a linear combination of separable basis functions.

5 Representation of Cast Shadow Effects in 3D Textures

The previous sections have shown how to formally analyze cast shadow effects, numerically simulated illumination basis functions, and experimentally validated the results. In this section, we make a first attempt at using this knowledge to efficiently represent lighting variability due to cast shadows in 3D textures.

In particular, our results indicate that a common set of illumination basis functions may be appropriate for many natural 3D textures. We will use an analytic basis motivated by the form of the illumination eigenmodes observed in the previous section. We use Equation 10, with the normalized basis functions written as

$$W_{mn}(\theta, \phi) = \sqrt{\frac{4m+3}{\pi}} P_{2m+1}(\cos \theta) a z_n(\phi), \quad (11)$$

where $a z_n(\phi)$ stands for $\cos n\phi$ or $\sin n\phi$, depending on whether n is plus or minus (and is $\sqrt{1/2}$ for $n = 0$), while P_{2m+1} are odd Legendre Polynomials.

This basis has some advantages over other possibilities such as spherical harmonics or Zernike polynomials for representing illumination over the hemisphere in 3D textures.

- The basis is specialized to the hemisphere, unlike spherical harmonics.
- Its form, as per Equations 10 and 11 is a simple product of 1D functions in θ and ϕ , simpler than Equation 9 for Zernike polynomials and spherical harmonics.
- For diffuse textures, due to visibility and shading effects, the intensity goes to 0 at grazing angles. These boundary conditions are automatically satisfied, since odd Legendre polynomials vanish at $\theta = \pi/2$ or $\cos \theta = 0$.
- Our basis seems consistent with numerical simulations and real experiments.

Figure 13 compares the resulting error with our basis to that for spherical harmonics, Zernike polynomials and the numerically computed optimal SVD basis for the sponge example. Note that the SVD basis performs best because it is tailored to the particular dataset and is by definition optimal. However, it requires prior knowledge of the data on a specific 3D texture, while we seek an analytic basis suitable for all 3D textures. These results demonstrate that our basis is competitive with other possibilities and can provide a good compact representation of measured illumination data in textures.

We demonstrate two simple applications of our analytic basis in Figure 14. In both cases, we use our basis to fit a function over the hemisphere. For 3D textures, this function is the illumination-dependence, fit separately at each pixel. The first application is to **compression**, wherein the original 6840 images are represented using 100 basis

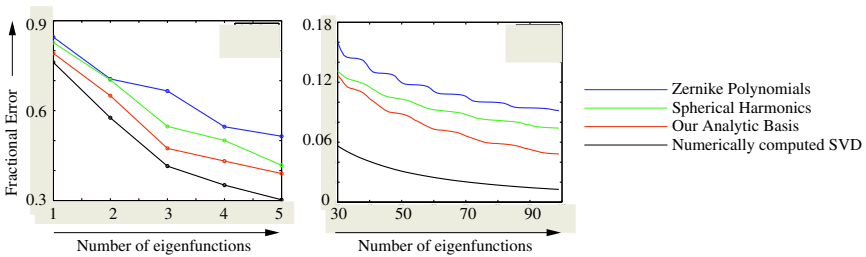


Fig. 13. Comparison of errors from different bases on sponge example (the left shows the first 6 terms, while the right shows larger numbers of terms). The SVD basis is tailored to this particular dataset and hence performs best; however it requires full prior knowledge.

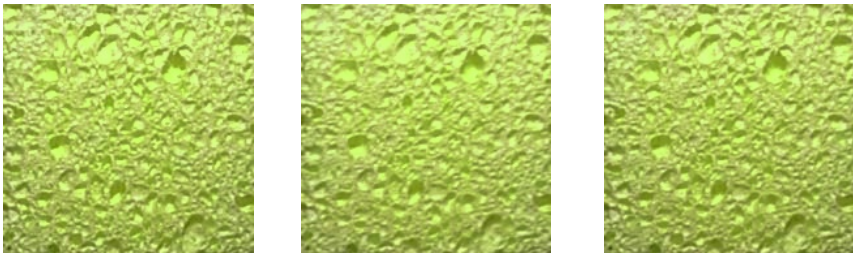


Fig. 14. On the left is one of the actual sponge images. In the middle is a reconstruction using 100 analytic basis functions, achieving a compression of 70:1. In the right, we reconstruct from a sparse set of 390 images, using our basis for interpolation and prediction. Note the subtle features of appearance, like accurate reconstruction of shadows, that are preserved.

function coefficients at each pixel. A compression of 70:1 is thus achieved, with only marginal loss in sharpness. This compression method was presented in [8], where a numerically computed SVD basis for textures sampled in both lighting and viewpoint was used. Using a standard analytic basis is simpler, and the same basis can now be used for all 3D textures. Further, note that once continuous basis functions have been fit, they can be evaluated for intermediate light source directions not in the original dataset. Our second application is to **interpolation** from a sparse sampling of 390 images. As shown in Figure 14, we are able to accurately reconstruct images not in the sparse dataset, potentially allowing for much faster acquisition times (an efficiency gain of 20 : 1 in this case), without sacrificing the resolution or quality of the final dataset.

It is important to discuss some limitations of our experiments and the basis proposed in equation 11. First, this is an initial experiment, and a full quantitative conclusion would require more validation on a variety of materials. Second, the basis functions in equation 11 are a good approximation to the eigenmodes derived from our numerical simulations, but the optimal basis will likely be somewhat different for specific shadowing configurations or 3D textures. Also, our basis functions are specialized to hemispherical illumination for macroscopically flat textures; the spherical harmonics or Zernike polynomials may be preferred in other applications. Another point concerns the use of our basis for representing general hemispherical functions. In particular, our basis functions go to 0 as $\theta = \pi/2$, which is appropriate for 3D textures, and similar to some spherical harmonic constructions over the hemisphere [22]. However, this makes it unsuitable for other applications, where we want a general hemispherical basis.

6 Conclusions

This paper formally analyzes cast shadows, showing that a simple Fourier signal-processing framework can be derived in many common cases. Our results indicate a theoretical link between cast shadows, and convolution formulae for irradiance and more general non-Lambertian materials [1,15,16]. This paper is also a first step in quantitatively understanding the effects of lighting in 3D textures, where cast shadows play a major role. In that context, we have derived new illumination basis functions over the hemisphere, which are simply a separable basis written as a product of odd Legendre polynomials and Fourier azimuthal functions.

Acknowledgements. We thank the reviewers for pointing out several important references we had missed. This work was supported in part by grants from the National Science Foundation (ITR #0085864 Interacting with the Visual world, and CCF # 0305322 Real-Time Visualization and Rendering of Complex Scenes) and Intel Corporation (Real-Time Interaction and Rendering with Complex Illumination and Materials).

References

- [1] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. In *ICCV 01*, pages 383–390, 2001.

- [2] K. Dana and S. Nayar. Histogram model for 3d textures. In *CVPR 98*, pages 618–624, 1998.
- [3] K. Dana, B. van Ginneken, S. Nayar, and J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, January 1999.
- [4] R. Epstein, P. Hallinan, and A. Yuille. 5 plus or minus 2 eigenimages suffice: An empirical investigation of low-dimensional lighting models. In *IEEE 95 Workshop Physics-Based Modeling in Computer Vision*, pages 108–116.
- [5] P. Hallinan. A low-dimensional representation of human faces for arbitrary lighting conditions. In *CVPR 94*, pages 995–999, 1994.
- [6] J. Koenderink, A. Doorn, K. Dana, and S. Nayar. Bidirectional reflection distribution function of thoroughly pitted surfaces. *IJCV*, 31(2/3):129–144, 1999.
- [7] J. Koenderink and A. van Doorn. Phenomenological description of bidirectional surface reflection. *JOSA A*, 15(11):2903–2912, 1998.
- [8] M. Koudelka, S. Magda, P. Belhumeur, and D. Kriegman. Acquisition, compression, and synthesis of bidirectional texture functions. In *ICCV 03 Workshop on Texture Analysis and Synthesis*, 2003.
- [9] T.M. MacRobert. *Spherical harmonics; an elementary treatise on harmonic functions, with applications*. Dover Publications, 1948.
- [10] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.
- [11] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *SIGGRAPH 01*, pages 519–528, 2001.
- [12] M. Oren and S. Nayar. Generalization of lambert’s reflectance model. In *SIGGRAPH 94*, pages 239–246, 1994.
- [13] R. Ramamoorthi. Analytic PCA construction for theoretical analysis of lighting variability in images of a lambertian object. *PAMI*, 24(10):1322–1333, 2002.
- [14] R. Ramamoorthi and P. Hanrahan. Analysis of planar light fields from homogeneous convex curved surfaces under distant illumination. In *SPIE Photonics West: Human Vision and Electronic Imaging VI*, pages 185–198, 2001.
- [15] R. Ramamoorthi and P. Hanrahan. On the relationship between radiance and irradiance: Determining the illumination from images of a convex lambertian object. *JOSA A*, 18(10):2448–2459, 2001.
- [16] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering. In *SIGGRAPH 01*, pages 117–128, 2001.
- [17] I. Sato, Y. Sato, and K. Ikeuchi. Illumination distribution from brightness in shadows: adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. In *ICCV 99*, pages 875–882, 1999.
- [18] P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics (SIGGRAPH 2002)*, 21(3):527–536, 2002.
- [19] C. Soler and F. Sillion. Fast calculation of soft shadow textures using convolution. In *SIGGRAPH 98*, pages 321–332.
- [20] P. Suen and G. Healey. Analyzing the bidirectional texture function. In *CVPR 98*, pages 753–758, 1998.
- [21] K. Torrance and E. Sparrow. Theory for off-specular reflection from roughened surfaces. *JOSA*, 57(9):1105–1114, 1967.
- [22] S. Westin, J. Arvo, and K. Torrance. Predicting reflectance functions from complex surfaces. In *SIGGRAPH 92*, pages 255–264, 1992.
- [23] L. Wolff, S. Nayar, and M. Oren. Improved diffuse reflection models for computer vision. *IJCV*, 30:55–71, 1998.

Surface Reconstruction by Propagating 3D Stereo Data in Multiple 2D Images

Gang Zeng¹, Sylvain Paris², Long Quan¹, and Maxime Lhuillier³

¹ Dep. of Computer Science, HKUST, Clear Water Bay, Kowloon, Hong Kong
`{zenggang, quan}@cs.ust.hk`

² ARTIS[†] / GRAVIR-IMAG, INRIA Rhône-Alpes, 38334 Saint Ismier, France
`sylvain.paris@imag.fr`

³ LASMEA, UMR CNRS 6602, Université Blaise-Pascal, 63177 Aubière, France
`Maxime.Lhuillier@lasmea.univ-bpclermont.fr`

Abstract. We present a novel approach to surface reconstruction from multiple images. The central idea is to explore the integration of both 3D stereo data and 2D calibrated images. This is motivated by the fact that only robust and accurate feature points that survived the geometry scrutiny of multiple images are reconstructed in space. The density insufficiency and the inevitable holes in the stereo data should be filled in by using information from multiple images. The idea is therefore to first construct small surface patches from stereo points, then to progressively propagate only reliable patches in their neighborhood from images into the whole surface using a best-first strategy. The problem reduces to searching for an optimal local surface patch going through a given set of stereo points from images. This constrained optimization for a surface patch could be handled by a local graph-cut that we develop. Real experiments demonstrate the usability and accuracy of the approach.

1 Introduction

Surface reconstruction from multiple images is one of the most challenging and fundamental problems of computer vision. Although effective for computing camera geometry, most recent approaches [9, 6] reconstruct only a 3D point cloud of the scene, whereas surface representations are indispensable for modelling and visualization applications. Surface reconstruction is a natural extension of the point-based geometric methods. Unfortunately, using 3D data from such a passive system in the same way as range scanner data is often insufficient for a direct surface reconstruction method, because the 3D points are sparse, irregularly distributed and missing in large areas. On the other hand, most image-based surface reconstruction approaches equally consider all surface points. They ignore the *feature points* although these points can be precisely matched between multiple images and therefore lead to accurate 3D locations. These shortcomings motivate us to develop a new approach to constructing a surface from stereo data [17], while using extra image information that is still available from a passive system.

[†] ARTIS is a research project in the GRAVIR/IMAG laboratory, a joint unit of CNRS, INPG, INRIA and UJF.

Surface reconstruction from 3D data. Surface reconstruction from scanned data is a traditional research topic. Szeliski et al. [27] use a particle-based model of deformable surfaces; Hoppe et al. [11] present a signed distance for implicit surfaces; Curless and Levoy [4] describe a volumetric method; and Amenta et al. [1] develop a set of computational geometry tools. Recently, Zhao et al. [32] develop a level-set method based on a variational method of minimizing a weighted minimal surface. Similar work is also developed by Whitaker [30] using a MAP framework. Surface reconstruction from depth data obtained from stereo systems is more challenging because the stereo data are usually much sparser and less regular. Fua [8] uses a system of particles to fit the stereo data. Kanade et al. [20] propose a deformable mesh representation to match the multiple dense stereo data. These methods that perform reconstruction by deforming an initial model or tracking discretized particles to fit the points are both topologically and numerically limited. Compared with our method, Hoff and Ahuja [10] and Szeliski and Golland [28] handle the data in an unordered way. Tang and Medioni [29] and Lee et al. [16] formulate the problem under a tensor voting framework.

Surface reconstruction from 2D images. Recently, several volumetric algorithms [25, 15, 5, 14] simultaneously reconstruct surfaces and obtain dense correspondences. The method of *space carving* or *voxel coloring* [25, 15] works directly on discretized 3D space, voxels, based on their image consistency and visibility. These methods are purely local and therefore rely either on numerous viewpoints or on textured surfaces to achieve satisfying results. Kolmogorov and Zabih [14], Roy [23], and Hishikawa and Geiger [12] propose direct discrete minimization formulations that are solved by graph-cuts. These approaches achieve disparity maps with accurate contours but limited depth precision. The latest improvement by Boykov and Kolmogorov [3] overcomes this point but is restricted to data segmentation. Paris et al. [22] also propose a continuous functional but is restricted to open surfaces. Faugeras and Keriven [5] propose a method implemented by level-sets, which is intrinsically a multiple view and naturally handles the topology and occlusion problems. However, it is not clear under what conditions their methods converge as the proposed functional seems non-convex.

Some ideas in our approach are inspired by these methods, but fundamentally these methods either solely operate on 3D data or on 2D data. Our approach is more similar to the work of Lhuillier and Quan [18], which integrates 3D and 2D data under a level-set framework and suffers from the same limitations. Our strategy to achieve this goal is to perform a propagation in 3D space starting from reliable feature points. The propagation is driven by image information to overcome the insufficiency of 3D data. Among possible types of image information, *cross-correlation* considers the local texture information and gives the surface location with great precision – especially its zero-mean normalized version (ZNCC) that is robust to lighting variations, but error matches occur when coherence constraints are not taken into consideration. We define a propagation technique as a local optimization to combine coherence constraints and image information. We make the surface grow patch by patch and we show that with an appropriate graph-cut technique, each patch is optimal under our hypotheses.

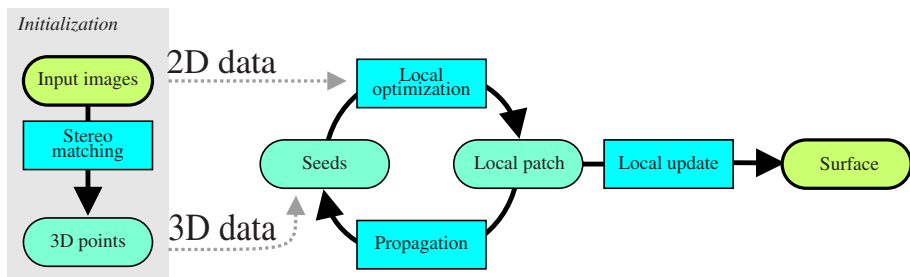


Fig. 1. Our framework is centered on the propagation loop that progressively extends the surface. Each iteration of the loop picks a seed, builds an optimal patch surrounding it, updates the surface and selects new seeds for further propagation. This process is initialized with 3D points computed by a stereoscopic method.

Our contributions. Our approach explores the integration of both stereo data and calibrated images. It fully exploits the feature points to start the reconstruction process to avoid potential ambiguities and build a precise surface. This is achieved through a new and innovative propagation framework controlled by 2D images. The graph-cut optimization is used locally to guarantee that the propagation is optimal under our hypotheses. Theoretical justifications are given for a formal understanding of the process and technical issues are exposed.

2 Problem Statement and Formulation

Given a set of calibrated images $\{C_j\}$, a set of stereo points $\mathcal{Q} = \{\mathbf{q}_i\}$ derived from the given images, the goal is to reconstruct a surface \mathcal{S} of the objects in the scene. The problem is different to surface reconstructions from multiple images as addressed in [5, 25, 15] in which only 2D images are used without any 3D information. It is also different to surface reconstructions from scanned 3D data without 2D image information [11, 27, 4, 8, 20, 32], as the 3D stereo data are often insufficient in density and accuracy for a traditional surface reconstruction.

Futhermore, most of the existing surface reconstruction techniques characterize their results by a global functional. However, Blake and Zisserman [2] demonstrate that their optimization scheme follows local rules. For instance, it is shown that a discontinuity has a local influence on the result and that this influence disappears beyond a given distance. Examining the global approach of Faugeras and Keriven [5] leads to the same remark: even if the initial formulation relies on a surface integral, the practical resolution is performed with local differential operators. This motivates our approach (Figure 1): the surface is built patch by patch, each one being locally optimal for a global criterion.

Since the patch is local, it is reasonable to establish the local coordinates system (x, y, z) where xy is the tangent plane to the object surface and z is the normal, to parameterize the patch as a single valued height field $z = h(x, y)$.

Then, we look for a surface patch $\{(x, y, h(x, y)) | (x, y) \in \mathcal{D}\}$ by minimizing a functional of type:

$$\iint_{\mathcal{D}} c(x, y, h(x, y)) dx dy, \quad (1)$$

where $c(x, y, h(x, y))$ is a cost function accounting for the consistency of the surface point $(x, y, h(x, y))$ in multiple images, for instance, either photo-consistency or cross-correlation.

We also require that the optimized surface patch goes through the existing 3D points if they do exist in the specified neighborhood. We are therefore looking for an interpolating surface through the given 3D points. That is, to minimize

$$\iint_{\mathcal{D}} c(x, y, h(x, y)) dx dy \quad \text{with} \quad h(x_i, y_i) = z_i, \quad (2)$$

where (x_i, y_i, z_i) is the local coordinate of a given 3D point \mathbf{q}_i . This functional can either be regularized as a minimal surface $\iint c(x, y, h(x, y)) ds$ leading to a level-set implementation [26, 5, 13]. We do not follow this path as it very often results in an over-smoothed surface [18] due to high order derivatives involved in the dynamic surface evolution. Instead, we apply a recent approach that reaches sharper results [22].

We use first derivatives as smoothing term

$$s(x, y, h(x, y)) = \alpha \left(\left| \frac{\partial h}{\partial x}(x, y) \right| + \left| \frac{\partial h}{\partial y}(x, y) \right| \right) \quad (3)$$

with α controlling the importance of $s(\cdot)$, to minimize the following functional

$$\iint_{\mathcal{D}} (c(x, y, h(x, y)) + s(x, y, h(x, y))) dx dy \quad \text{with} \quad h(x_i, y_i) = z_i. \quad (4)$$

The advantage of this formulation is that this continuous functional can be discretized, then optimized by a graph-cut algorithm. This is an adaption of the graph-cut approach, with a simplified smoothing term $s(\cdot)$ to a local environment that we call a *local graph-cut*. The choice of $c(\cdot)$ is purely independent of the general framework. Recently researchers mainly focus on *photo-consistency* and *cross-correlation*. In this paper we present an implementation of the propagation framework with $c(\cdot) = g(zncc(\cdot))$ with $g(\cdot)$ being a decreasing function like $x \mapsto (1 - x)$ to fit the minimization goal.

This local formulation relies on the surface orientation to choose the parameterization of the local patch, the normal direction of a given 3D point has to be estimated from its neighborhood. We define a seed point as a couple $(\mathbf{p}_i, \mathbf{n}_i)$ where \mathbf{p}_i is the 3D position and \mathbf{n}_i the surface normal direction at \mathbf{p}_i . This is dependent on the reconstruction algorithms used, which is further discussed in Section 4. The above analysis finally leads to a surface that optimally interpolates the given set of 3D stereo points by 2D image information defined in the functional (4).

3 Algorithm Description

We propose here an algorithm to apply the previously discussed framework. This algorithm follows the general organization presented in Figure 1. First, using the technique discussed in Section 4, 3D stereo points are computed. The local surface orientation at these points is estimated to attach with the points to form the initial seeds. Then the information is propagated from the most reliable data to grow the surface.

3.1 Initialization from 3D Stereo Points

To start the algorithm, we initialize the list of seeds, from which the surface is propagated gradually. A high number of 3D points are robustly computed from the given set of images by stereo and bundle-adjustment methods (Section 4). Figure 4 shows these points for different data sets. Compared with standard sparse points, these points are highly redundant and well distributed in several parts. This redundancy makes it possible to evaluate the surface orientation. It is important to address here that these input points are regarded as a “black box” without assuming any property. Other preprocessing methods are also possible as long as the surface orientation can be estimated at each given position.

For each 3D point \mathbf{p}_i , the surface orientation \mathbf{n}_i is provided by the symmetric 3×3 positive semi-definite matrix $\sum_{\mathbf{y} \in \mathcal{B}_r(\mathbf{p}_i) \cap \mathcal{Q}} (\mathbf{y} - \mathbf{p}_i) \otimes (\mathbf{y} - \mathbf{p}_i)$, where $\mathcal{B}_r(\mathbf{p}_i)$ denotes the ball of radius r . Among the eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ respectively associated to the eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$, we choose \mathbf{n}_i to be either \mathbf{v}_3 or $-\mathbf{v}_3$. The sign depends on the cameras used to reconstruct \mathbf{p}_i . The confidence is indicated by the ratio between the lowest eigenvalue and the larger ones.

$\mathcal{B}_r(\mathbf{p}_i) \cap \mathcal{Q}$ may contain very few points, baffling the orientation estimation. And in dense regions, a large radius results in an over-smoothed estimation whereas a small radius makes the estimation sensitive to noise. Therefore r is defined as a function of \mathbf{p}_i : in dense regions, r is fixed to a reference value r_{dense} representing the minimum scale. In the diluted regions, the radius is increased so that $\mathcal{B}_r(\mathbf{p}_i)$ contains at least k 3D stereo points. From many experiments, a good compromise is to define r_{dense} to be the radius of local patches and k to be 15-20.

3.2 Propagation Loop

Starting from the seed list initialized with the 3D stereo points, this step makes the surface grow until it reaches the final result. Each seed is handled one by one to generate a surface patch and create new seeds by the local graph-cut technique. It is divided into the 3 following issues:

1. The selection of the next seed from the current seed list.
2. The generation of an optimal patch from this seed.
3. The creation of the new seeds on this patch to add in the list.

3.3 Selection of the Next Seed

To select a new seed (\mathbf{p}, \mathbf{n}) for propagation, we need a criterion Π to evaluate how “good for propagation” a seed is. Once we have such a criterion, we follow a classical *best-first strategy* to ensure that the most reliable seed is picked each time. This choice directly drives the propagation because it indicates where the growing regions are. It is our global control over the surface reconstruction.

First of all, the initial seeds (*i.e.* 3D stereo points) are regarded as the reliable 3D points on the surface. Therefore, they are always selected before the generated seeds. The algorithm ends when there is no seed left in the list.

Selection criterion for 3D stereo points. The criterion of the stereo points is defined as the confidence of the orientation estimation described in Section 3.1. This confidence is related to the local planarity of the stereo points and therefore estimates the accuracy of the normal computation. This leads to $\Pi = \frac{\lambda_2}{\lambda_3}$. The visibility of a stereo point is given by the cameras used to reconstruct it.

Selection criterion for generated seeds. For a generated seed, we use the ZNCC correlation score Z by its two most front-facing cameras, since a strong match gives a high confidence. This strategy ensures that the surface grows from the part which is more likely to be precise and robust. Thus: $\Pi = Z$. Correct visibility is computed thanks to this propagation criterion. If the criterion is computed from occluded cameras, the local textures in both images does not match and the ZNCC value is low. Therefore a seed without occlusion is processed before a seed with occlusion. The occluded parts “wait” until other parts are reconstructed. The visibility of the processed seed is classically determined by the current propagated surface using a ray-tracing technique.

3.4 Generation of an Optimal Patch from a Given Seed

Given a seed (\mathbf{p}, \mathbf{n}) with its visibility, a patch is grown in its neighborhood to extend the existing surface. Since it is a local representation of the surface, it is parameterized by a height field $z = h(x, y)$ relatively to the tangent plane of the surface. Therefore, for the local coordinate (x, y, z) , the x and y axes are arbitrary chosen orthogonally to \mathbf{n} , and the z axis is parallel to \mathbf{n} with the coordinate origin at \mathbf{p} . A surface patch is defined by $\{(x, y, h(x, y)) | (x, y) \in \mathcal{D}\}$ that is minimal for functional (4). This patch is also enforced to pass through the stereo points and the previously computed surface.

The graph-cut technique described in [22] is then applied. This technique reaches an exact minimum of the functional (4) up to any arbitrary discretization. This involves a graph whose cuts represent the set of all possible surfaces $z = h(x, y)$. Thus the classical max-flow problem [7] is equivalent to an exhaustive search leading to a minimal solution of (4). Moreover, it allows us to constrain $h(x, y)$ to a specified z -range.

To use this technique, the local domain is discretized as a regular rectangular grid $\{X_1, \dots, X_{n_x}\} \times \{Y_1, \dots, Y_{n_y}\} \times \{Z_1, \dots, Z_{n_z}\}$ separated by $\Delta x, \Delta y, \Delta z$. The functional (4) is discretized into the following form:

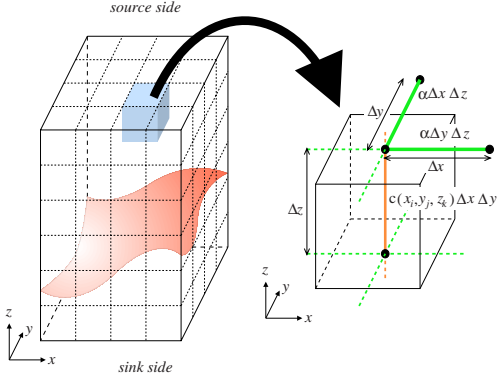


Fig. 2. Correspondence between the graph and the voxel $(x, y, h(x, y))$.

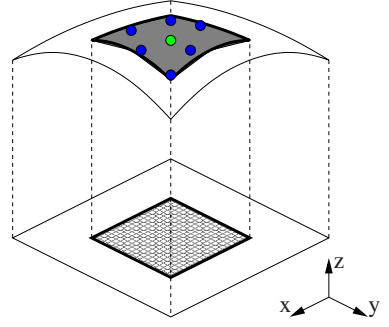


Fig. 3. Only the center part of the optimal solution is actually used to extend the existing surface (shaded area).

$$\sum_i \sum_j (c(X_i, Y_j, h(X_i, Y_j)) \Delta x \Delta y + |h(X_{i+1}, Y_j) - h(X_i, Y_j)| \Delta y$$

$$+ |h(X_i, Y_{j+1}) - h(X_i, Y_j)| \Delta x) \quad \text{with the constraint} \quad h(x_i, y_i) = z_i. \quad (5)$$

For any $(x, y) \in \mathcal{D}$ where the surface has been already computed (including the 3D stereo points), the h value is restricted to the only corresponding z value. Then for each remaining (x, y, z) position, a ZNCC value is computed using the two most fronto-parallel visible cameras to limit the perspective distortion. These ZNCC values are transformed into $c(\cdot)$, resulting in a dense 3D sampling.

Based on the above discrete form (5), a graph is embedded into a 3D grid superimposed on the voxels with correspondence shown in Figure 2. Finally, this graph-flow problem is solved and the resulting minimum cut is equivalent to the optimal patch. Graph design and proof of optimization are in [22]. The confidence of the patch is given by the quantity of the maximum flow F . A smaller value gives a higher confidence, while the large enough value indicates that this patch should be discarded.

Even if the patch is trustworthy, the borders of the patch might not be suitable for the final surface. The border points have a truncated neighborhood and potentially ignore some neighboring data. To avoid this caveat, only the center part of the optimal solution is kept (see Figure 3). The discarded border only provides a complete neighborhood to the center part. Finally, this center part is used to extend the existing surface.

3.5 Creation of the New Seeds to Add in the List

We have grown a surface patch for the selected seed in the previous section. To continue the propagation, new seeds are created from this patch. These new seeds will be selected later to drive further propagation. The location of the new seeds is driven by several aspects.

Patch quality. First of all, the quantity of the maximum flow F indicates the confidence of the optimal patch. With low enough confidence, the surface patch is discarded and no seed is created.

Match quality. A point with a high ZNCC value Z is more likely to provide a robust starting point for further propagation.

Surface regularity. A singular point does not represent accurate properties of the patch. Using the principal curvatures κ_1 and κ_2 , points with high curvature $K = \kappa_1^2 + \kappa_2^2$ are therefore to be avoided.

Propagation efficiency. To ensure a faster propagation, distant points are preferred. This relies on the distance D between the patch center and the potential new seeds.

A value Λ is computed for each potential location of a new seed to represent its appropriateness relative to these objectives.

$$\Lambda = \frac{Z^{\gamma(Z)} \cdot D^{\gamma(D)}}{F^{\gamma(F)} \cdot K^{\gamma(K)}} \quad (6)$$

where $\gamma(\cdot)$ are non-negative weights to balance the different criteria. In our algorithm, we use $\gamma(Z) = \gamma(D) = \gamma(F) = \gamma(K) = 1$, while further study is needed to evaluate the importance of each criterion.

The number of new seeds created is inspired by the triangle mesh configuration. From the Euler property, the average number of neighbors of a vertex is 6 and the average angular distance between two neighbors is $\frac{\pi}{3}$. Thus, the directions of the new seeds in relation to the patch center are selected so that the angular distance between two neighbor seeds lies in $[\frac{2\pi}{5}, \frac{2\pi}{7}]$. In each direction, the location \mathbf{p}' with the highest Λ is selected and the normal \mathbf{n}' at \mathbf{p}' is attached to form a new seed.

4 Implementation and Experiments

4.1 Acquisition of 3D Stereo Data

3D stereo data can be obtained using different approaches. A traditional stereo rig [21, 24] could deliver quite dense points, but it only gives a small part of the object. Automatically merging partial stereo data into a complete model is not easy, except that multiple stereo rigs are calibrated off-line. We choose a more general reconstruction method from an uncalibrated sequence [6, 9, 31]. Our quasi-dense implementation [17] is similar to the standard uncalibrated approaches with the main difference that we use a much denser set of points re-sampled from the disparity map instead of points of interest. To model a complete object, we usually make a full turn around the object by capturing about 30 images to compute the geometry of the sequence.

4.2 Efficiency Consideration

To select a seed to propagate, we use selection criterion Π to extend the surface from its most robust regions (Section 3.3). However, it may cause inefficiency, because the neighboring surface may have already been created from other seeds.

Therefore, the *area criterion* A is added to define $\tilde{\Pi} = A \cdot \Pi$. The definition of A represents the efficiency of the patch generation. It is based on the total patch area a_P and the covered area a_C : $A = 1 - \frac{a_C}{a_P}$. Since this value evolves during the process, it is stored with the seed and updated in a “lazy” way: When the best seed is selected by $\tilde{\Pi}$, the corresponding A is updated according to the current surface. If the value changes to a non-zero value (it always decreases because the surface is growing), $\tilde{\Pi}$ is updated and the seed is put back in the list. If it changes to zero, the seed is discarded because it would provide no surface extension. This case mainly occurs for 3D stereo points which are redundant. Otherwise (*i.e.* A does not change), the seed is selected for further processing.

4.3 Representation of the Growing Surface with a Distance Field

We use a distance field to register the growing surface. The surface is represented by the signed distance function d in its *narrow band*, which is Euclidean distance with the sign indicating either inside or outside. When a surface patch is generated, the distance field is updated in its narrow band. This results in the merger of this patch with its neighboring ones. Finally, the zero set $\mathcal{Z}(d)$ is our estimation for \mathcal{S} , which is extracted by the *marching cube* method [19].

4.4 Experimental Results

Three typical modelling examples are shown in Figures 4. We usually acquire about 30 images with a hand-held camera around the objects. The first “toy” example is used to examine the correctness and robustness of our algorithm. The “toy” is difficult since fur is traditionally hard for surface reconstruction. We have made a tradeoff between the local features and the orientation robustness based on the local coherence. As we can see from the reconstructed shape, our algorithm handles occlusion correctly. Two “face” examples illustrate the accuracy of our algorithm. One face has more texture, while the other has less. The details around the eyes, the noses and the ears can be clearly seen thanks to the exploitation of the feature points. The shape is also satisfactory in the other regions, such as the foreheads and the cheeks because of the new propagation technique. It is important to address that the final shape optimally interpolates the given set of 3D stereo points by 2D image information defined in the functional (4).

The results in the above two face examples are of almost same quality. We may notice that the 3D feature points are good surface points, but their quantity is not an important factor for reconstruction. Since the derived 3D points are highly redundant, actually many of the points provide limited information. Future study is needed for the quantity and the positions of the initial seeds.

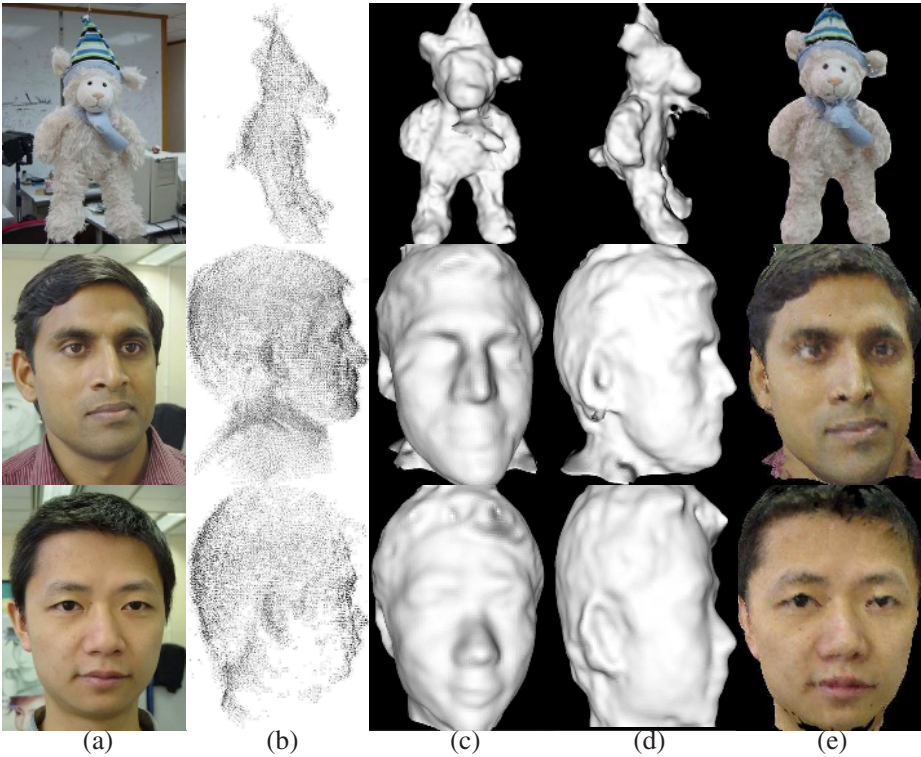


Fig. 4. Each row shows the results for one example: (a) One of the input images; (b) Reconstructed 3D points; (c,d) Surface shape at two different viewpoints; (e) Surface shape with color.

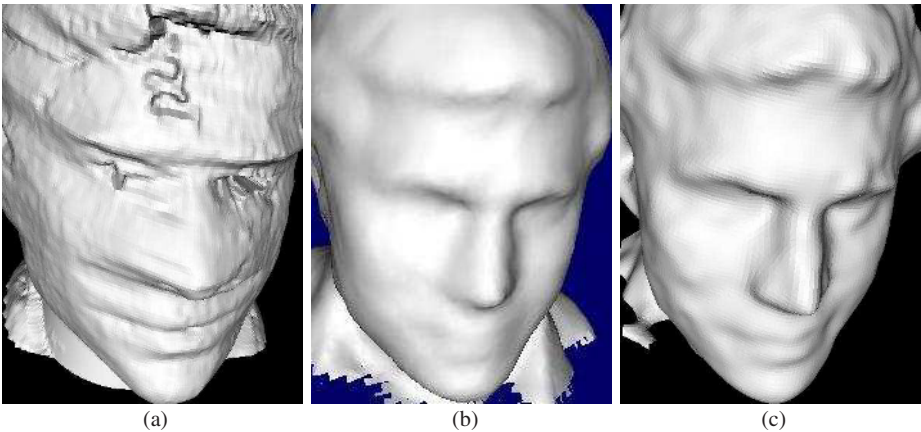


Fig. 5. Comparative results: (a) Space carving only gives a rough estimation; (b) Level set method yields an over-smoothed result; (c) The propagation gives the most detailed surface geometry.

Figure 5 shows a comparative study among the space carving method [15], the level set method [18] and our propagation approach. The space carving method only uses image information, and misses a lot of details on the nose, the hair and the ears due to its photo-consistency criterion, which often yields an over-estimation on the intensity-homogeneous regions. The level set method and our approach are both based on stereo points and images. However the level set method tends to over-smooth the surface by losing many geometric details.

5 Conclusions

We have proposed a novel approach for surface reconstruction by propagating 3D stereo data in multiple 2D images. It is based on the fact that the feature points can be accurately and robustly determined. These points give important information for surface reconstruction. On the other hand, they are not sufficient to generate the whole surface representation. The major motivation of this study is to improve the insufficiency of 3D stereo data by using original 2D images.

Our strategy to achieve this goal is to perform a propagation in a 3D space starting from reliable feature points. We have introduced a new functional (4) integrating both stereo data points and image information. The surface grows patch by patch from the most reliable regions by the local graph cut technique. Finally, the shape optimally interpolates the given set of 3D stereo points by 2D image information defined in the functional (4). This approach have been extensively tested on real sequences and very convincing results have been shown.

Acknowledgements. This project is supported by the Hong Kong RGC grant HKUST 6188/02E. Also thanks to Yichen Wei and Ajay Kumar for allowing us to use their head images.

References

1. N. Amenta, S. Choi, and R. Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 2001.
2. A. Blake and A. Zisserman. Visual reconstruction. *MIT Press*, 1987.
3. Y. Boykov and V. Kolmogorov. Computing Geodesics and Minimal Surfaces via Graph Cuts. *Int. Conf. on Computer Vision*, 2003.
4. B. Curless and M. Levoy. A volumetric method for building complex models from range images. *SIGGRAPH*, 1996.
5. O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. *European Conf. on Computer Vision*, 1998.
6. O. Faugeras, Q. Luong, and T. Papadopoulos. Geometry of Multiple Images. *MIT Press*, 2001.
7. L. Ford and D. Fulkerson. Flows in Networks. *Princeton University Press*, 1962.
8. P. Fua. From multiple stereo views to multiple 3d surfaces. *Int. Journal of Computer Vision*, 1997.
9. R.I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. *CU Press*, 2000.

10. W. Hoff, N. Ahuja Surfaces from Stereo: Integrating Feature Matching, Disparity Estimation, and Contour Detection. *Trans. on Pattern Analysis and Machine Intelligence*, 1989.
11. H. Hoppe, T. Deroose, T. Duchamp, J. McDonaft, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 1992.
12. H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. *European Conf. on Computer Vision*, 1998.
13. R. Kimmel 3D Shape Reconstruction from Autostereograms and Stereo. *Journal of Visual Communication and Image Representation*, 2002.
14. V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. *European Conf. on Computer Vision*, 2002.
15. K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *Int. Conf. on Computer Vision*, 1999.
16. M.S. Lee, G. Medioni and P. Mordohai. Inference of Segmented Overlapping Surfaces from Binocular Stereo. *Trans. on Pattern Analysis and Machine Intelligence*, 2002.
17. M. Lhuillier and L. Quan. Quasi-dense reconstruction from image sequence. *European Conf. on Computer Vision*, 2002.
18. M. Lhuillier and L. Quan. Surface Reconstruction by Integrating 3D and 2D Data of Multiple Views. *Int. Conf. on Computer Vision*, 2003.
19. W.E. Lorensen and H.E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH*, 1987.
20. P.J. Narayanan, P.W. Rander and T. Kanade. Constructing Virtual Worlds using Dense Stereo. *European Conf. on Computer Vision*, 1998.
21. M. Okutomi and T. Kanade. A multiple-baseline stereo. *Trans. on Pattern Analysis and Machine Intelligence*, 1993.
22. S. Paris and F. Sillion and L. Quan. A Surface Reconstruction Method Using Global Graph Cut Optimization *Asian Conf. on Computer Vision*, 2004.
23. S. Roy. Stereo without epipolar lines : A maximum-flow formulation. *Int. Journal of Computer Vision*, 1999.
24. H.S. Sawhney, H. Tao and R. Kumar. A global matching framework for stereo computation. *Int. Conf. on Computer Vision*, 2001.
25. S.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. *Computer Vision and Pattern Recognition*, 1997.
26. J.A. Sethian. Level-set methods and fast marching methods. *CU Press*, 1999.
27. R. Szeliski, D. Tonnesen, and D. Terzopoulos. Modelling surfaces of arbitrary topology with dynamic particles. *Computer Vision and Pattern Recognition*, 1993.
28. R. Szeliski and P. Golland. Stereo matching with transparency and matting. *Int. Journal of Computer Vision*, 1999.
29. C.K. Tang and G. Medioni. Curvature-augmented tensor voting for shape inference from noisy 3d data. *Trans. on Pattern Analysis and Machine Intelligence*, 2002.
30. R. Whitaker. A level-set approach to 3d reconstruction from range data. *Int. Journal of Computer Vision*, 1998.
31. Z. Zhang and Y. Shan. A progressive scheme for stereo matching. *SMILE'2, Workshop of European Conf. on Computer Vision*, 2000.
32. H.K. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and non-parametric shape reconstruction from unorganized data using a variational level set method. *Computer Vision and Image Understanding*, 2000.

Visibility Analysis and Sensor Planning in Dynamic Environments

Anurag Mittal¹ and Larry S. Davis²

¹ Real-Time Vision and Modeling, Siemens Corporate Research, Princeton, NJ 08540.
anurag@scr.siemens.com

² Computer Science Department, University of Maryland, College Park, MD 20742.
lsd@cs.umd.edu

Abstract. We analyze visibility from static sensors in a dynamic scene with moving obstacles (people). Such analysis is considered in a probabilistic sense in the context of multiple sensors, so that visibility from even one sensor might be sufficient. Additionally, we analyze worst-case scenarios for high-security areas where targets are non-cooperative. Such visibility analysis provides important performance characterization of multi-camera systems. Furthermore, maximization of visibility in a given region of interest yields the optimum number and placement of cameras in the scene. Our analysis has applications in surveillance - manual or automated - and can be utilized for sensor planning in places like museums, shopping malls, subway stations and parking lots. We present several example scenes - simulated and real - for which interesting camera configurations were obtained using the formal analysis developed in the paper.

1 Introduction

We present a method for sensor planning that is able to determine the required number and placement of static cameras (sensors) in a dynamic scene. Such analysis has previously been presented for the case of static scenes where the constraints and obstacles are static. However, in many applications, apart from these static constraints, there exists occlusion due to dynamic objects (people) in the scene. In this paper, we incorporate these dynamic visibility constraints into the sensor planning task. These constraints are analyzed in a probabilistic sense in the context of multiple sensors. Furthermore, we develop tools for analyzing worst-case visibility scenarios that are more meaningful for high-security areas where targets are non-cooperative.

Our analysis is useful for both manned and automated vision systems. In manned systems where security personnel are looking at the video stream, it is essential that the personnel have visibility of the people in the scene. In automated systems, where advanced algorithms are used to detect and track multiple people from multiple cameras, our analysis can be used to place the cameras in an optimum configuration.

Automated Multi-camera vision systems have been developed using a wide range of camera arrangements. For better stereo matching, some systems[1] use closely-spaced cameras. Others [2,3] adopt the opposite arrangement of widely separated cameras for maximum visibility. Others [4] use a hybrid approach. Still others [5,6,7], use multiple cameras for the main purpose of increasing the field of view. In all these systems, there is

a need for analyzing the camera arrangement for optimum placement. In many cases, our method can be utilized without any alteration. In systems that have additional algorithmic requirements (e.g. stereo matching), further constraints - *hard* or *soft* - can be specified so that the optimum camera configuration satisfies (*hard*) and is optimum (*soft*) w.r.t. these additional constraints.

In addition to providing the optimum configuration, our analysis can provide a *gold standard* for evaluating the performance of these systems under the chosen configuration. This is because our analysis provides the theoretical limit of detectability. No algorithm can surpass such a limit since the data is missing from the images. Thus, one can determine as to how much of the error in a system is due to missing data, and how much of it is due to the chosen algorithm.

Sensor planning has been researched quite extensively, especially in the robotics community, and there are several different variations depending on the application. One set of methods use an active camera mounted on a robot. The objective then is to move the camera to the best location in the next view based on the information captured up to now. These methods are called next view planning [8,9,10]. Another set of methods obtain a model (either 2D or 3D) of a scene by optimum movement of the camera [11,12]. Such model acquisition imposes certain constraints on the camera positions, and satisfaction of these constraints guarantees optimum and stable acquisition.

Methods that are directly related to ours are those that determine the location of static cameras so as to obtain the best views of a scene. This problem was originally considered in the computational geometry literature as the art-gallery problem [13]. The solutions in this domain utilize simple 2D or 3D scene models and simple assumptions on the cameras and occlusion in order to develop theoretical results and efficient algorithms to determine good sensor configurations (although the NP-hard nature of the problem typically necessitates an approximate solution). Several researchers [14,15,16,17] have studied and incorporated more complex constraints based on several factors not limited to (1) resolution, (2) focus, (3) field of view, (4) visibility, (5) view angle, and (6) prohibited regions. In addition to these “static” constraints, there exist additional “visibility” constraints imposed by the presence of dynamic obstacles. Such constraints have not been analyzed earlier and their incorporation into the sensor planning task constitutes the novel aspect of our work.

The paper is organized as follows. Section 2 develops the theoretical framework for estimating the probability of visibility of an object at a given location in a scene for a certain configuration of sensors. Section 3 introduces some deterministic tools to analyze worst-case visibility scenarios. Section 4 describes the development of a cost function and its minimization in order to perform sensor planning in complex environments. Section 5 concludes the paper with some simulated and real experiments.

2 Probabilistic Visibility Analysis

In this section, we analyze probabilistically the visibility constraints in a multi-camera setting. Specifically, we develop tools for evaluating the probability of visibility of an object from at least one sensor. Since this probability varies across space, this probability is recovered for each possible object position.

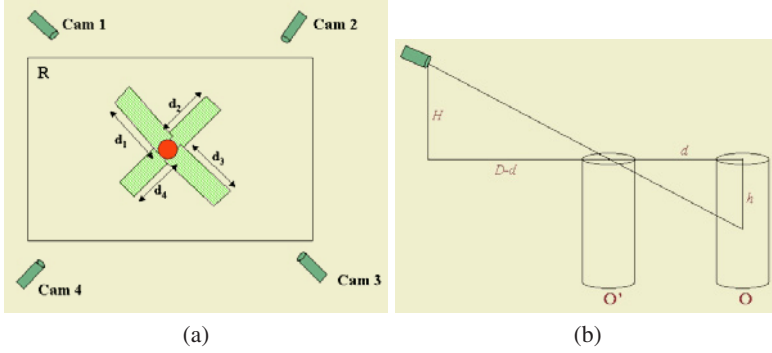


Fig. 1. (a) Scene Geometry used for stochastic reasoning. (b) The distance up to which an object can occlude another object is proportional to its distance from the sensor.

2.1 Visibility from at Least One Sensor

Assume that we have a region \mathcal{R} of area A observed by n sensors [Fig. 1 (a)]. Let \mathcal{E}_i be the event that a target object \mathcal{O} at location \mathcal{L} is visible from sensor i . The probability that \mathcal{O} is visible from at least one sensor can be expressed mathematically as the union $P(\bigcup_{i=1}^n \mathcal{E}_i)$ of these events, and it can be expanded using the inclusion-exclusion principle as:

$$P(\bigcup_i \mathcal{E}_i) = \sum_{\forall i} P(\mathcal{E}_i) - \sum_{i < j} P(\mathcal{E}_i \cap \mathcal{E}_j) + \cdots + (-1)^{n+1} P(\bigcap_i \mathcal{E}_i) \quad (1)$$

The motivation for this expansion is that it is easier to compute the terms on the RHS (right hand side) compared to the one on the LHS.

In order to facilitate the introduction of the approach to be followed in computing $P(\bigcup_{i=1}^n \mathcal{E}_i)$, we consider the specific case of objects moving on a ground plane. The objects are also assumed to have the same horizontal profile at each height. Examples of such objects include cylinders, cubes, cuboids, and square prisms, and can adequately describe the objects of interest in many applications such as people detection and tracking. Let the area of their projection onto the ground plane be A_{ob} . Furthermore, we assume that the sensors are placed at some known heights H_i from this plane. Also, we define visibility to mean that the center line of the object (corresponding to the centroid in the horizontal profile) is visible for at least some length h from the top of the object (in people tracking, this might correspond to viewing the face).

A useful quantity can be defined for the objects by considering the projection of the object in a particular direction. We then define r as the average, over different directions, of the maximum distance from the centroid to the projected object points. For e.g., for cylinders, r is the radius; for square prism with side $2s$, $r = \frac{1}{\pi/4} \int_0^{\pi/4} s \cos \theta d\theta = 2\sqrt{2}s/\pi$. The quantity r will be useful in calculating the average occluding region of an object. Furthermore, it can easily be shown that the distance d_i up to which an object can occlude another object is proportional to its distance D_i from sensor i [Fig. 1 (b)]. Mathematically,

$$d_i = (D_i - d_i)\mu_i = D_i \frac{\mu_i}{\mu_i + 1}, \quad \text{where} \quad \mu_i = \frac{h}{H_i} \quad (2)$$

Fixed Number of Objects. In order to develop the analysis, we start with the case of a fixed number k of objects in the scene under the assumption that they are located randomly and uniformly in region \mathcal{R} . This will be extended to the more general case of object densities in subsequent sections.

Under this assumption, we first estimate $P(\mathcal{E}_i)$, which refers to the probability that none of the k objects is present in the region of occlusion \mathcal{R}_i^o for camera i . Assuming that all object orientations are equally likely¹, one may approximate the area of this region of occlusion as $A_i^o \approx d_i(2r)$. Then, the probability for a single object to *not* be present in this region of occlusion is $\left(1 - \frac{A_i^o}{A}\right)$. Since there are k objects in the scene located independently of each other, the probability that none of them is present in the region of occlusion is $\left(1 - \frac{A_i^o}{A}\right)^k$. Thus:

$$P(\mathcal{E}_i) = \left(1 - \frac{A_i^o}{A}\right)^k \quad (3)$$

In order to provide this formulation, we have neglected the fact that two objects cannot overlap each other. In order to incorporate this condition, we observe that the $(j+1)$ -th object has a possible area of only $A - jA_{ob}$ available to it². Thus, Equation 3 can be refined as

$$P(\mathcal{E}_i) = \prod_{j=0}^{k-1} \left(1 - \frac{A_i^o}{A - jA_{ob}}\right) \quad (4)$$

This analysis can be generalized to other terms in Equation 1. The probability that the object is visible from all of the sensors in a specified set $(i_1, i_2 \dots i_m)$ can be determined as:

$$P\left(\bigcap_{i \in (i_1, i_2, \dots, i_m)} \mathcal{E}_i\right) = \prod_{j=0}^{k-1} \left(1 - \frac{A_{(i_1, i_2, \dots, i_m)}^o}{A - jA_{ob}}\right) \quad (5)$$

where $A_{(i_1, \dots, i_m)}^o$ is the area of the combined region of occlusion $\mathcal{R}_{(i_1, \dots, i_m)}^o$ for the sensor set (i_1, \dots, i_m) formed by the “geometric” union of the regions of occlusion $\mathcal{R}_{i_p}^o$ for the sensors in this set, i.e. $\mathcal{R}_{(i_1, \dots, i_m)}^o = \bigcup_{p=1}^m \mathcal{R}_{i_p}^o$.

Uniform Object Density. A fixed assumption on the number of objects in a region is clearly inadequate. A more realistic assumption is that the objects have a certain density of occupancy. First, we consider the case of uniform object density in the region. This

¹ It is possible to perform the analysis by integration over different object orientations. However, for ease of understanding, we will use this approximation.

² The prohibited area is in fact larger. For example, for cylindrical objects, another object cannot be placed anywhere within a circle of radius $2r$ (rather than r) without intersecting the object. For simplicity and ease of understanding, we redefine A_{ob} as the area “covered” by the object. This is the area of the prohibited region and may be approximated as four times the actual area of the object.

will be extended to the more general case of non-uniform object density in the next section. The uniform density case can be treated as a generalization of the “ k objects” case introduced in the previous section. To this end, we increase k and the area A proportionately such that

$$k = \lambda A \quad (6)$$

where a constant object density λ is assumed. Equation 5 can then be written as

$$P\left(\bigcap_{i \in (i_1, \dots, i_m)} \mathcal{E}_i\right) = \lim_{k \rightarrow \infty} \prod_{j=0}^{k-1} \left(1 - \frac{A_{(i_1, \dots, i_m)}^o}{k/\lambda - jA_{ob}}\right) \quad (7)$$

We define:

$$a = \frac{1}{\lambda A_{(i_1, \dots, i_m)}^o}, \quad b = \frac{A_{ob}}{A_{(i_1, \dots, i_m)}^o} \quad (8)$$

Here, a captures the effect of the presence of objects and b is a *correction* to such effect due to the finite object size. Then, we obtain:

$$P\left(\bigcap_{i \in (i_1, \dots, i_m)} \mathcal{E}_i\right) = \lim_{k \rightarrow \infty} \prod_{j=0}^{k-1} \left(1 - \frac{1}{ka - jb}\right) \quad (9)$$

Combining terms for j and $k - j$, we get

$$\begin{aligned} & \left(1 - \frac{1}{ka - jb}\right) \left(1 - \frac{1}{ka - (k-j)b}\right) \\ &= \frac{k^2 a^2 - k^2 ab - 2ka + j(k-j)b^2 + bk + 1}{k^2 a^2 - k^2 ab + j(k-j)b^2} \end{aligned}$$

Assuming $a \gg b$ (i.e. the object density λ is much smaller than $1/A_{ob}$, the object density if the area is fully packed), we can neglect terms involving b^2 . Then, the above term can be written as

$$\approx \left(1 - \frac{1}{k} \left(\frac{2a - b}{a^2 - ab}\right)\right)$$

There are $k/2$ such terms in Equation 9. Therefore,

$$P\left(\bigcap_{i \in (i_1, \dots, i_m)} \mathcal{E}_i\right) \approx \lim_{k \rightarrow \infty} \left(1 - \frac{1}{k} \left(\frac{2a - b}{a^2 - ab}\right)\right)^{k/2}$$

Using the identity $\lim_{x \rightarrow \infty} \left(1 + \frac{1}{x}\right)^x = e$, we get

$$P\left(\bigcap_{i \in (i_1, \dots, i_m)} \mathcal{E}_i\right) \approx e^{-\frac{2a-b}{2a(a-b)}} \quad (10)$$

Non-uniform Object Density. In general, the object density (λ) is a function of the location. For example, the object density near a door might be higher. Moreover, the presence of an object at a location influences the object density nearby since objects tend to appear in groups. We can integrate both of these influences on the object density with the help of a conditional density function $\lambda(\mathbf{x}_c|\mathbf{x}_O)$ that might be available to us. This density function gives the density at location \mathbf{x}_c given that visibility is being calculated at location \mathbf{x}_O . Thus, this function is able to capture the effect that the presence of the object at location \mathbf{x}_O has on the density nearby³.

In order to develop the formulation for the case of non-uniform density, we note that the $(j + 1)$ -th object has a region available to it that is \mathcal{R} minus the region occupied by the j previous objects. This object is located in this “available” region according to the density function $\lambda(\cdot)$. The probability for this object to be present in the region of occlusion $R_{(i_1, \dots, i_m)}^o$ can then be calculated as the ratio of the average number of people present in the region of occlusion to the average number of people in the available region. Thus, one can write:

$$P\left(\bigcap_{i \in (i_1, \dots, i_m)} \mathcal{E}_i\right) = \lim_{k \rightarrow \infty} \prod_{j=0}^{k-1} \left(1 - \frac{\int_{\mathcal{R}_{(i_1, \dots, i_m)}^o} \lambda(\mathbf{x}_c|\mathbf{x}_O) d\mathbf{x}_c}{\int_{\mathcal{R}-\mathcal{R}_{ob}^j} \lambda(\mathbf{x}_c|\mathbf{x}_O) d\mathbf{x}_c}\right) \quad (11)$$

where \mathcal{R}_{ob}^j is the region occupied by the previous j objects. Since the previous j objects are located randomly in \mathcal{R} , one can simplify:

$$\int_{\mathcal{R}-\mathcal{R}_{ob}^j} \lambda(\mathbf{x}_c|\mathbf{x}_O) d\mathbf{x}_c = \lambda_{avg}(A - jA_{ob})$$

where λ_{avg} is the average object density in the region. Using this simplification in Equation 11 and noting that $\lambda_{avg}A = k$, we obtain:

$$P\left(\bigcap_{i \in (i_1, \dots, i_m)} \mathcal{E}_i\right) = \lim_{k \rightarrow \infty} \prod_{j=0}^{k-1} \left(1 - \frac{\int_{\mathcal{R}_{(i_1, \dots, i_m)}^o} \lambda(\mathbf{x}_c|\mathbf{x}_O) d\mathbf{x}_c}{k - j \cdot \lambda_{avg} \cdot A_{ob}}\right) \quad (12)$$

Defining:

$$a = \frac{1}{\int_{\mathcal{R}_{(i_1, \dots, i_m)}^o} \lambda(\mathbf{x}_c|\mathbf{x}_O) d\mathbf{x}_c}, \quad b = \frac{A_{ob} \cdot \lambda_{avg}}{\int_{\mathcal{R}_{(i_1, \dots, i_m)}^o} \lambda(\mathbf{x}_c|\mathbf{x}_O) d\mathbf{x}_c}, \quad (13)$$

Equation 12 may again be put in the form of Equation 9. As before, this may be simplified to obtain the expression in Equation 10.

2.2 Visibility from Multiple Sensors

In many applications, it is desirable to view an object from more than one sensor. Stereo reconstruction/depth recovery is an example where the requirement of visibility from at

³ Such formulation only captures the first-order effect of the presence of an object. While higher order effects due to the presence of multiple objects can be considered, they are likely to be small.

least two sensors is to be satisfied. In order to evaluate the probability of visibility from at least two sensors, one can evaluate:

$$P\left(\bigcup_{(i < j)} (\mathcal{E}_i \cap \mathcal{E}_j)\right) \quad (14)$$

This term can be expanded exactly like Equation 1 treating each term $(\mathcal{E}_i \cap \mathcal{E}_j)$ as a single entity. All the terms on the RHS will then have only intersections in them which are easy to compute using the formulation developed in the previous sections.

2.3 Additional Constraints

Other “static” constraints also affect the view of a particular camera. Therefore, the visibility probability needs to be calculated after incorporating these additional constraints. This is easily achieved in our scheme since the visibility constraints are analyzed at individual locations and additional constraints can also be verified at these locations. The constraints that have been incorporated in our system include:

1. **FIELD OF VIEW:** Cameras have a limited field of view. At each location, it can be verified whether that location is within the field of view of a particular camera.
2. **OBSTACLES:** Fixed *high* obstacles like pillars cause occlusions in certain areas. From a given location, it needs to be determined whether any obstacle blocks the view of a particular camera.
3. **PROHIBITED AREAS:** There might also exist prohibited areas where people are not able to walk. An example of such an area is a desk. These areas have a positive effect on the visibility in their vicinity since it is not possible for obstructing objects to be present within such regions.
4. **RESOLUTION:** The resolution of an object in an image reduces as the object moves further away from the camera. Therefore, meaningful observations are possible only up to a certain distance from the camera. It can easily be verified whether the location is within a certain “resolution distance” from the camera.
5. **ALGORITHMIC CONSTRAINTS:** There are several algorithmic constraints that may exist. For example, stereo matching across two (or more) cameras imposes a constraint on the maximum distortion of the view that can occur from one camera to the other. This constraint can be expressed in terms of the angular separation between the camera centers from the point of view of the object. It can be easily be verified whether this constraint is satisfied at a particular location.
6. **VIEWING ANGLE:** An additional constraint exists for the maximum angle α_{max} at which the observation of an object is meaningful. Such observation can be the basis for performing some other tasks like object recognition. This constraint translates into a constraint on the minimum distance from the sensor that an object must be. This minimum distance guarantees the angle of observation to be smaller than α_{max} .

The analysis presented so far is probabilistic and provides “average” answers. In high security areas, worst-case analysis might be more appropriate. Such analysis will be presented in the next section.

3 Worst-Case Visibility Analysis

In this section, we present some simple results for location-specific limitations of a given system in the worst-case. This analysis provides conditions that guarantee visibility regardless of object configuration and enables sensor placement such that such conditions are satisfied in a given region of interest. Since the analysis is quite simple, we will only briefly describe these results. We propose:

Theorem 1. *Suppose there is an object \mathcal{O} at location \mathcal{L} . If there are k point objects in the vicinity of \mathcal{O} , and n sensors have visibility of location \mathcal{L} , then $n > k + m - 1$ is the necessary and sufficient condition to guarantee visibility for \mathcal{O} from at least m sensors.*

Proof. (a) *Necessary:* Suppose $n \leq k + m - 1$. Place $p = \min(k, n)$ objects such that each obstructs one sensor. The number of sensors having a clear view of the object are then equal to $n - p$ which is less than m (follows easily from the condition $n \leq k + m - 1$). (b) *Sufficient :* Suppose $n > k + m - 1$. \mathcal{O} has n lines of sight to the sensors, k of which are possibly obstructed by other objects. Therefore, by the extended pigeon-hole principle, there must be at least $n - k \geq m$ sensors viewing \mathcal{O} .

This result holds for point objects only. It can be extended to finite objects if certain assumptions are made. One can assume a flat world scenario where the objects and the sensors are in 2D. Also assume that we are given a point of interest in the object such that object visibility is defined as the visibility of this point of interest. This point can be defined arbitrarily. Let us also define an angle α as the maximum angle that any object can subtend at the point of interest of any other object. For example, for identical cylinders with the center as the point of interest, $\alpha = 60^\circ$. For identical square prisms, $\alpha = 90^\circ$. Under these assumptions, the above result holds if we take n to be the number of sensors that have visibility of location \mathcal{L} such that the angular separation between any two sensors, from the point of view of \mathcal{L} , is at least α . Also, n must be less than $2\pi/\alpha$ since it is not possible to place $n > 2\pi/\alpha$ sensors such that there is an angular separation of at least α between them.

For a given camera configuration, one can determine the number of cameras that each location of interest has visibility to. This will yield the maximum number of people that can be present in the vicinity of the person and still guarantee visibility for him.

4 Sensor Planning

The visibility analysis presented in section 2 yields a function $p_s(\mathbf{x})$, that refers to the probability that an object located at location \mathbf{x} is visible from at least one of the sensors that have the parameter vector \mathbf{s} . Such parameter vector may include, for instance, the location, viewing direction and zoom of each camera. Given such a function, one can define a suitable *cost* function in order to evaluate a given set of sensor parameters. Such sensor parameters may be further constrained due to other factors. For instance, there typically exists a physical limitation on the positioning of the cameras (walls, ceilings etc.). The sensor planning problem can then be formulated as a problem of constrained optimization of the cost function. Such optimization will yield the optimum sensor parameters according to the specified cost function.

4.1 The Cost Function

Several cost functions may be considered. Based on deterministic visibility analysis, one can consider a simple cost function that sums, over the region of interest \mathcal{R}_i , the number $N(\mathbf{x})$ of cameras that a location \mathbf{x} has visibility to:

$$C(\mathbf{s}) = - \sum_{\mathbf{x} \in \mathcal{R}_i} N(\mathbf{x}) \quad (15)$$

Using probabilistic analysis, one may define a cost function that minimizes the maximum occlusion probability in the region:

$$C(\mathbf{s}) = \max_{\mathbf{x} \in \mathcal{R}_i} (1 - p_{\mathbf{s}}(\mathbf{x}))$$

Another cost function, and perhaps the most reasonable one in many situations, is to define the cost as the negative of the average number of visible people in a given region of interest:

$$C(\mathbf{s}) = - \int_{\mathcal{R}_i} \lambda(\mathbf{x}) p_{\mathbf{s}}(\mathbf{x}) d\mathbf{x} \quad (16)$$

This cost function has been utilized for obtaining the results in this paper.

It is also possible to integrate other constraints into the cost function. For instance, some of the constraints in section 2.3 may be specified as *soft* constraints rather than *hard* constraints (for e.g. resolution, viewing angle and algorithmic constraints). According to the application, any arbitrary function of the constraints may be considered:

$$C(\mathbf{s}) = f(c_1, \dots, c_J, \lambda(), \mathcal{R}_i)$$

where $c_j, j = 1 \dots J$ are the different constraints to be satisfied.

4.2 Minimization of the Cost Function

The cost function defined by Equation 16 (as also other suitable ones) is non-linear and it can be shown that it is not differentiable. Furthermore, in most non-trivial cases, it has multiple local minima and possibly multiple global minima. Fig. 2 illustrates the cost function for the scene shown in Fig. 4 (a). where, for illustration purposes, only two of the nine parameters have been varied. Even in this two dimensional space, there are two global minima and several local minima. Furthermore, the gradient is zero in some regions.

Due to these characteristics of the cost function, it is not possible to minimize it using simple gradient-based methods that can only find the local minimum of a well-behaved “convex” function. Global minimization methods that can deal with complex cost functions are necessary [18]. Simulated Annealing and Genetic Algorithms are two classes of algorithms that may be considered. The nature of the cost function suggests that either of these two algorithms should provide an acceptable solution[19]. For our experiments, we implemented a simulated annealing scheme using a highly sophisticated simulated re-annealing software ASA developed by L. Ingber [20].

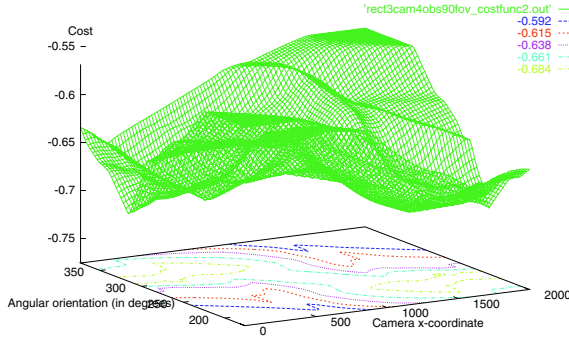


Fig. 2. The Cost Function for the scene in Fig. [4 (a)] where, for illustration purposes, only the x-coordinate and direction of the second camera have been varied.

Using this algorithm, we were able to obtain extremely good sensor configurations in a reasonable amount of time (5min - a couple of hours on a Pentium IV 2.2GHz PC, depending on the desired accuracy of the result, the number of dimensions of the search space and complexity of the scene). For low dimensional spaces (< 4), where it was feasible to verify the results using full search, it was found that the algorithm quickly converged to a global minimum. For moderate dimensions of the search space (< 8), the algorithm was again able to obtain the optimum solution, but only after some time. Although the optimality of the solution could not be verified by full search, we assumed such solution to be optimum since running the algorithm several times from different starting points and different annealing parameters did not alter the final solution. For very high dimensional spaces (> 8), although the algorithm provided “good” solutions very quickly, it took several hours to converge to the best one. Some of the “optimal” solutions thus obtained will be illustrated in the next section.

5 Simulations and Experiments

We have proposed a stochastic algorithm for recovering the optimal sensor configuration with respect to certain visibility requirements. In order to validate the proposed method, we provide results of the algorithm for various scenes, synthetic and real.

5.1 Synthetic Experiments

In all the synthetic examples we consider next, we take a rectangular room of size 10mX20m. The sensors were restricted to be mounted $H = 2.5$ m above the ground and have a field of view of 90° . We use a uniform object density $\lambda = 1m^{-2}$, object height = 150cm, object radius $r=15$ cm, minimum visibility height $h=50$ cm and maximum visibility angle $\alpha_{max} = 45^\circ$. The illustrations shown are visibility maps scaled such that

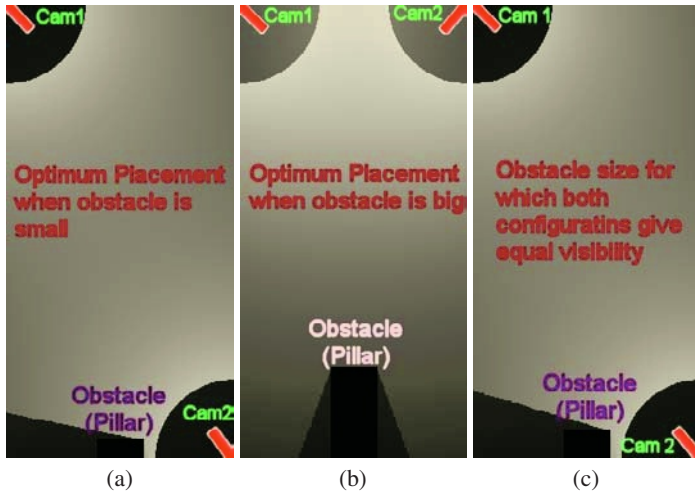


Fig. 3. Illustration of the effect of scene geometry on sensor placement. Optimum configuration when (a): obstacle size is small. (b): obstacle size is big. (c): obstacle size is such that both configurations are equally good.

$[0,1]$ maps onto $[0,255]$, thus creating a gray scale image. Brighter regions represent higher visibility. Note how the visibility decreases as we move away from a camera due to an increase in the distance of occlusion d_i .

Fig. 3 illustrates the effect that an obstacle can have on camera placement. Using a maximum of two cameras having a field of view of 90° , the first configuration [a] was found to be optimum when the obstacle size was small ($<60\text{cm}$). Configuration [b] was optimum when the object size was big ($>60\text{cm}$). For the object size shown in configuration [c] ($\sim 60\text{cm}$), both configurations were equally good. Note that, in both configurations, all locations are visible from at least one camera. Therefore, current methods based solely on analysis of static obstacles would not be able to distinguish between the two.

Fig. 4 illustrates how the camera specifications can significantly alter the optimum sensor configuration. Notice that the scene has both obstacles and prohibited areas. With three available cameras, configuration [a] was found to be optimum when the cameras have only 90° field of view but are able to “see” up to 25m. With the same resolution, configuration [b] is optimum if the cameras have a 360° field of view (Omni-Camera). If the resolution is lower so that cameras can “see” only up to 10m, configuration [c] is optimum.

Fig. 5 illustrates the effect of different optimization criteria. With the other assumptions the same as above, configuration [a] was found to be optimum when the worst case analysis was utilized [Eq. 15]. On the other hand, a uniform object density assumption [Eq. 16] yielded configuration [b] as the optimum one. When an assumption of variable object densities was utilized such that the density is highest near the door and decreases linearly with the distance from it [d], configuration [c] was found to be the best. Note that a higher object density near the door leads to a repositioning of the cameras such that they can better capture this region.

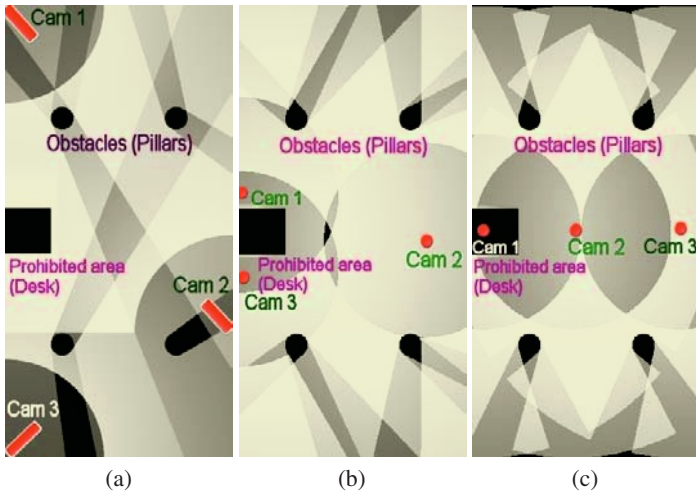


Fig. 4. Illustration of the effect of different camera specifications. With a uniform density assumption, the optimum configuration when the cameras have (a): field of view of 90° and resolution up to 25m, (b): 360° field of view (Omni-Camera), and resolution up to 25m, (c): 360° field of view, but resolution only up to 10m.

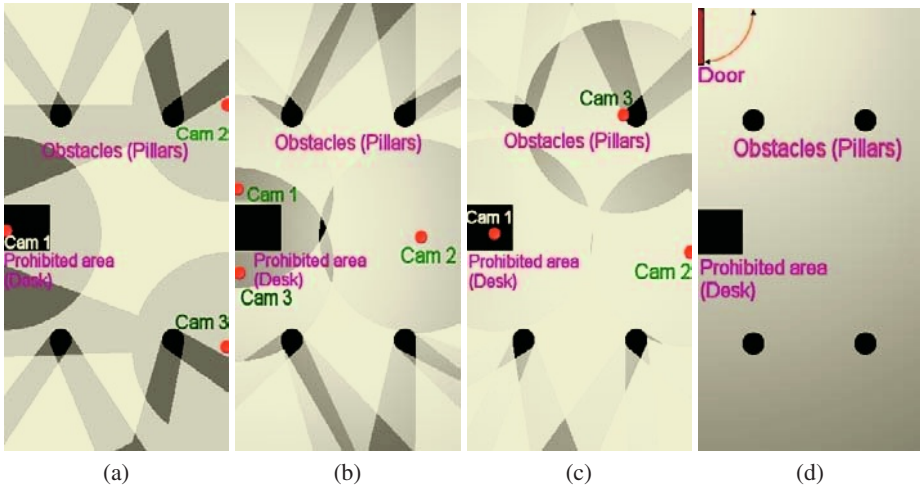


Fig. 5. Illustration of the effect of different optimization criteria. Optimum configuration for: (a): worst-case analysis [Eq. 15], (b): uniform density case [Eq. 16], (c): variable density case [Eq. 16] for the object density shown in (d).

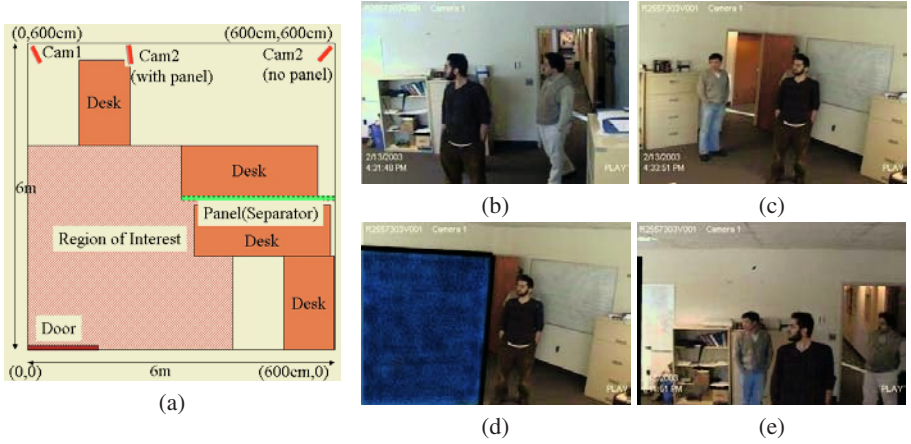


Fig. 6. (a) Plan view of a room used for a real experiment. (b) and (c) are the views from the optimum camera locations when there is no panel (obstacle). Note that, of the three people in the scene, one person is occluded in each view. However, all of them are visible from at least one of the views. Image (d) shows the view from the second camera in the presence of the panel. Now, one person is not visible in any view. To improve visibility, the second camera is moved to (180, 600). The view from this new location is shown in (e), where all people are visible again.

5.2 Analysis of a Real Scene

We now present analysis of sensor placement for a real office room. The structure of the room is illustrated in Fig. 6 (a). We used the following parameters - uniform density $\lambda = 0.25m^{-2}$, object height = 170cm, $r = 23cm$, $h = 40cm$, and $\alpha_{max} = 60^\circ$. The cameras available to us had a field of view of 45° and needed to be mounted on the ceiling which is 2.5m high. In order to view people's face as they enter the room, we further restricted the cameras to be placed on the wall facing the door. We first consider the case when there is no panel (separator). If only one camera is available, the best placement was found to be at location (600,600) at an angle of 135° (measured clockwise from the positive x-axis). If two cameras are available, the best configuration consists of one camera at (0,600) at an angle of 67.5° and the other camera at (600, 600) at an angle of 132° . Figures 6 (b) and (c) show the views from the cameras.

Next, we place a thin panel at location (300, 300) - (600, 300). The optimum configuration of two cameras consists of a camera at (0,600) at an angle of 67.5° (same as before) and the other camera at (180, 600) at an angle of 88° . Figures 6 (d) & (e) show the views from the original and new location of the second camera.

6 Conclusion

We have presented two methods for evaluation of visibility given a certain configuration of sensors in a scene. The first one evaluates the visibility probabilistically assuming a density function for the occluding objects. The second method evaluates worst-case scenarios and is able to provide conditions that would guarantee visibility regardless of

object configuration. Apart from obtaining important performance characterization of multi-sensor systems, such analysis was further used for sensor planning by optimization of an appropriate cost function. The algorithm was tested on several synthetic and real scenes, and in many cases, the configurations obtained were quite interesting and non-intuitive. The method has applications in surveillance and can be utilized for sensor planning in places like museums, shopping malls, subway stations and parking lots. Future work includes specification of more complex cost functions, investigation of more efficient methods for optimization of the cost function and better estimation of visibility probability by considering the effect of interaction between objects.

Acknowledgments. We would like to thank Nikos Paragios for help in improving the presentation of the paper, and Visvanathan Ramesh for helpful discussions on the topic.

References

- [1] Darrell, T., Demirdjian, D., Checka, N., Felzenszwalb, P.: Plan-view trajectory estimation with dense stereo background models. In: ICCV, Vancouver, Canada (2001) II: 628–635
- [2] Mittal, A., Davis, L.: M₂tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *IJCV* **51** (2003) 189–203
- [3] Khan, S., Javed, O., Rasheed, Z., Shah, M.: Human tracking in multiple cameras. In: ICCV, Vancouver, Canada (2001) I: 331–336
- [4] Collins, R., Lipton, A., Fujiyoshi, H., Kanade, T.: Algorithms for cooperative multi-sensor surveillance. *Proceedings of the IEEE* **89** (2001) 1456–1477
- [5] Stauffer, C., Grimson, W.: Learning patterns of activity using real-time tracking. *PAMI* **22** (2000) 747–757
- [6] Kettner, V., Zabih, R.: Counting people from multiple cameras. In: ICMCS. (1999) II:253–259
- [7] Cai, Q., Aggarwal, J.: Tracking human motion in structured environments using a distributed-camera system. *PAMI* **21** (1999) 1241–1247
- [8] Miura, J., Ikeuchi, K.: Task-oriented generation of visual sensing strategies. In: ICCV, Boston, MA (1995) 1106–1113
- [9] Ye, Y., Tsotsos, J.: Sensor planning for 3d object search. *CVIU* **73** (1999) 145–168
- [10] Pito, R.: A solution to the next best view problem for automated surface acquisition. *PAMI* **21** (1999) 1016–1030
- [11] Kutulakos, K., Dyer, C.: Recovering shape by purposive viewpoint adjustment. *IJCV* **12** (1994) 113–136
- [12] Cameron, A., Durrant-Whyte, H.: A bayesian approach to optimal sensor placement. *IJRR* **9** (1990) 70–88
- [13] O'Rourke, J.: *Art Gallery Theorems and Algorithms*. Oxford University Press (1987)
- [14] Cowan, C.K., Kovesi, P.: Automatic sensor placement from vision task requirements. *PAMI* **10** (1988) 407–416
- [15] Reed, M.K., Allen, P.K.: Constraint-based sensor planning for scene modeling. *PAMI* **22** (2000) 1460–1467
- [16] Tarabanis, K., Tsai, R., Kaul, A.: Computing occlusion-free viewpoints. *PAMI* **18** (1996) 279–292
- [17] Yi, S., Haralick, R., Shapiro, L.: Optimal sensor and light-source positioning for machine vision. *CVIU* **61** (1995) 122–137

- [18] Shang, Y.: Global Search Methods for Solving Nonlinear Optimization Problems. PhD thesis, University of Illinois at Urbana-Champaign (1997)
- [19] Duda, R., Hart, P., Stork, D.: Pattern Classification. John Wiley and Sons (2001)
- [20] Ingber, L.: Very fast simulated re-annealing. *Mathematical Computer Modeling* **12** (1989) 967–973

Camera Calibration from the Quasi-affine Invariance of Two Parallel Circles

Yihong Wu, Haijiang Zhu, Zhanyi Hu, and Fuchao Wu

National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, P.O. Box 2728, Beijing 100080, P.R. China
{yhwu,hjzhu,huzy,fcwu}@nlpr.ia.ac.cn

Abstract. In this paper, a new camera calibration algorithm is proposed, which is from the quasi-affine invariance of two parallel circles. Two parallel circles here mean two circles in one plane, or in two parallel planes. They are quite common in our life.

Between two parallel circles and their images under a perspective projection, we set up a quasi-affine invariance. Especially, if their images under a perspective projection are separate, we find out an interesting distribution of the images and the virtual intersections of the images, and prove that it is a quasi-affine invariance.

The quasi-affine invariance is very useful which is applied to identify the images of circular points. After the images of the circular points are identified, linear equations on the intrinsic parameters are established, from which a camera calibration algorithm is proposed. We perform both simulated and real experiments to verify it. The results validate this method and show its accuracy and robustness. Compared with the methods in the past literatures, the advantages of this calibration method are: it is from parallel circles with minimal number; it is simple by virtue of the proposed quasi-affine invariance; it does not need any matching.

Excepting its application on camera calibration, the proposed quasi-affine invariance can also be used to remove the ambiguity of recovering the geometry of single axis motions by conic fitting method in [8] and [9]. In the two literatures, three conics are needed to remove the ambiguity of their method. While, two conics are enough to remove it if the two conics are separate and the quasi-affine invariance proposed by us is taken into account.

1 Introduction

Camera calibration is an important task in computer vision whose aim is to estimate the camera parameters. Usually, camera self-calibration techniques without prior knowledge on camera parameters are nonlinear [4], [13], [15]. It can be linearized if some scene information is taken into account during the process of calibration. Therefore, it has been appearing a lot of calibration methods using scene constraints [2], [3], [5], [10], [11], [12], [14], [18], [19], [20], [23], [24], [25]. Usually, the used information in the scene is parallels, orthogonality, or

the known angles, circles and their centers, concentric conics et al. For example, in [14], the images of circular points are determined when there is a circle with several diameters in the scene, then the linear constraints on the intrinsic parameters are set up. In [2], by using the parallel and orthogonal properties of the scene, the constraints on the projective matrix are given. Parallelepipeds with some known angles and length ratios of the sides are assumed existed, then from them the equations on the intrinsic parameters are established in [20]. [25] presents a calibration method using one-dimensional objects.

Our idea in this paper is also to use the scene information to find the constraints on the intrinsic parameters of cameras. Two circles in one plane or in two parallel planes, called two parallel circles, are assumed to be in the scene, and then a quasi-affine invariance of them is found. Based on the invariance, camera calibration is investigated, and a new algorithm is proposed. Compared with the previous methods, this method has the following advantages: it is from parallel circles with minimal number; it is simple by virtue of the proposed quasi-affine invariance; it does not need any matching.

The parallel circles are quite common in our life, and then this calibration method can be applied. It can also be used to solve the ambiguity for recovering the geometry of single axis motions in [8], [9]. The two literatures have shown that the geometry of single axis motion can be recovered given at least two conic loci consisting of corresponding image points over multiple views. If the two conics are separate or enclosing, the recovery has a two fold ambiguity, the ambiguity is removed by using three conics in the literatures. In fact, if the two conics are separate, it is enough to remove the ambiguity only from the two conics by taking into account the quasi-affine invariance presented in this paper.

On the other hand, in [16], Quan gave the invariants of two space conics. When the two conics are parallel circles, this invariants cannot be set up, but a quasi-affine invariance proposed in this paper indeed exists. Actually, the imaging process of a pinhole camera is quasi-affine [6], [7], the proposed quasi-affine invariance is very useful.

The paper is organized as follows. Section 2 is some preliminaries. Section 3 uses a quasi-affine invariance of two parallel circles to establish the equations on the camera intrinsic parameters, and gives a linear algorithm for calibrating a camera from these equations. Then, the invariance and algorithm are validated from both simulated and real experiments in Section 4. Conclusions and acknowledgements are remarked in Section 5 and 6 respectively.

2 Preliminaries

In this paper, " \approx " denotes the equality up to a scale, a capital bold letter denotes a matrix or a 3D homogeneous coordinates, a small bold letter denotes a 2D homogeneous coordinates.

Definition 1. *If two circles in space are in one plane, or in two parallel planes respectively, we call them two parallel circles.*

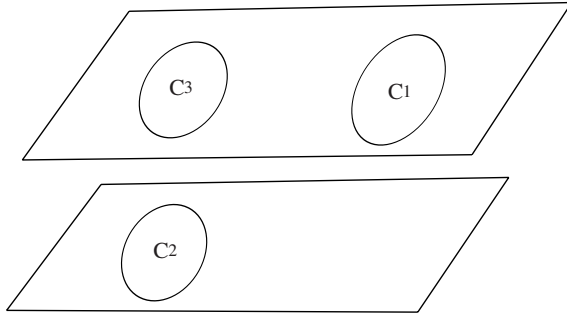


Fig. 1. Parallel circles. \mathbf{C}_1 and \mathbf{C}_3 are coplanar, \mathbf{C}_2 is in the plane parallel to the plane containing \mathbf{C}_1 and \mathbf{C}_3 . Any two of them are two parallel circles

See Fig. 1, \mathbf{C}_1 , \mathbf{C}_2 , \mathbf{C}_3 are parallel circles each other.

Under a pinhole camera, a point \mathbf{X} in space is projected to a point \mathbf{x} in the image by:

$$\mathbf{x} \approx \mathbf{K}[\mathbf{R}, \mathbf{t}]\mathbf{X}, \quad (1)$$

where \mathbf{K} is the 3×3 matrix of camera intrinsic parameters, \mathbf{R} is a 3×3 rotation matrix, \mathbf{t} is a 3D translation vector. The goal of calibrating a camera is to find \mathbf{K} from images.

The absolute conic consists of points $\mathbf{X} = (X_1, X_2, X_3, 0)$ at infinity such that:

$$X_1^2 + X_2^2 + X_3^2 = 0, \quad \text{or,} \quad \mathbf{X}^T \mathbf{X} = 0,$$

and its image ω is:

$$\mathbf{x}^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{x} = 0. \quad (2)$$

If some points on ω can be inferred from image, the equations on the intrinsic parameters can be set up by (2). If the number of these equations is enough, the intrinsic parameters will be determined. In the following, we are to find the points on ω by using two parallel circles in the scene.

Some preliminaries on projective geometry are needed, the readers can refer to the details in [17]. Every real plane other than the plane at infinity, denoted by P , intersects the plane at infinity at a real line, called the line at infinity of P , denoted by L_0 . L_0 intersects the absolute conic at a pair of conjugate complex points, called the circular points of P . Every circle in P passes through the circular points of P . Let \mathbf{C}_1 and \mathbf{C}_2 be two parallel circles, P_1 and P_2 be the parallel planes containing them. Because P_1 and P_2 have the same line at infinity, they have the same pair of circular points. Therefore, \mathbf{C}_1 and \mathbf{C}_2 pass through the same pair of circular points, they and the absolute conic form a coaxial conic system at the two circular points (A coaxial conic system means a set of conics through two fixed points).

A quasi-affine transformation lies part way between a projective and affine transformation, which preserves the convex hull of a set of points, and the relative

positions of some points and lines in a plane, or the relative positions of some points and planes in 3D space. For the details, see [7] or Chapter 20 in [6].

3 New Calibration Method from the Quasi-affine Invariance of Two Parallel Circles

Under a pinhole camera, a circle is projected to a conic. Moreover, because \mathbf{K} , \mathbf{R} , \mathbf{t} in (1) are real, a real point is projected to a real point, and a pair of conjugate complex is projected to a pair of conjugate complex. So the images of a pair of circular points must still be a pair of conjugate complex.

If there are three or more than three parallel circles in the scene, then from their images, the images of a pair of circular points can be uniquely determined without any ambiguity by solving for the intersection points of the three image conics [8], [21]. If there are only two ones in the scene, denoted by \mathbf{C}_1 , \mathbf{C}_2 , whose images are denoted by \mathbf{c}_1 , \mathbf{c}_2 , whether the images of a pair of circular points can be uniquely determined or not depends on the relative positions of \mathbf{c}_1 and \mathbf{c}_2 . The equations for \mathbf{c}_1 , \mathbf{c}_2 are two quadric equations, the number of their common solutions over complex field is four with multiplicity. If there are real solutions among these four ones, or \mathbf{c}_1 and \mathbf{c}_2 have real intersections, then there is a unique pair of conjugate complex among these four solutions, which must be the images of the pair of circular points. If \mathbf{c}_1 and \mathbf{c}_2 have no real intersection, these four solutions are two pairs of conjugate complex. Which pair is the images of the circular points? We will discuss it in the following.

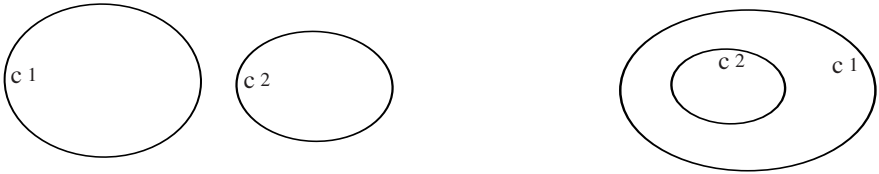


Fig. 2. Two cases that \mathbf{c}_1 and \mathbf{c}_2 have no real intersection: the left side is the separate case; the right side is the enclosing case

If the relative positions of the camera and circles in the scene are in general, or, the circles lie entirely in front of the camera, the images of these circles are ellipses. From now, we always regard that the circles in the scene are entirely in front of the camera. Then, when \mathbf{c}_1 and \mathbf{c}_2 have no real intersection, there are two cases as shown in Fig. 2, one case is that \mathbf{c}_1 and \mathbf{c}_2 separate; another case is that \mathbf{c}_1 and \mathbf{c}_2 enclose. For the enclosing case, we can not distinguish the images of the circular points between the two pairs of conjugate complex intersections of \mathbf{c}_1 and \mathbf{c}_2 [21]. While, for the separate case, we can distinguish them by a quasi-affine invariance.

Firstly, a lemma with respect to two coplanar circles is needed. In order to distinguish notations of two coplanar circles from the above notations \mathbf{C}_1 and \mathbf{C}_2 of two parallel circles, we denote two coplanar circles as \mathbf{C}_1 and \mathbf{C}_3 .

Lemma 1. *If \mathbf{C}_1 and \mathbf{C}_3 are two coplanar separate circles, their homogeneous equations have two pairs of conjugate complex common solutions. We connect the two points in each pair of the conjugate complex common solutions, then obtain two real lines, called the associated lines of \mathbf{C}_1 and \mathbf{C}_3 . One of the two associated lines lies between \mathbf{C}_1 and \mathbf{C}_3 , and the other one, which is the line at infinity passing through the circular points, does not lie between \mathbf{C}_1 and \mathbf{C}_3 as shown in Fig. 3.*

The proof of Lemma 1 is given in Appendix.

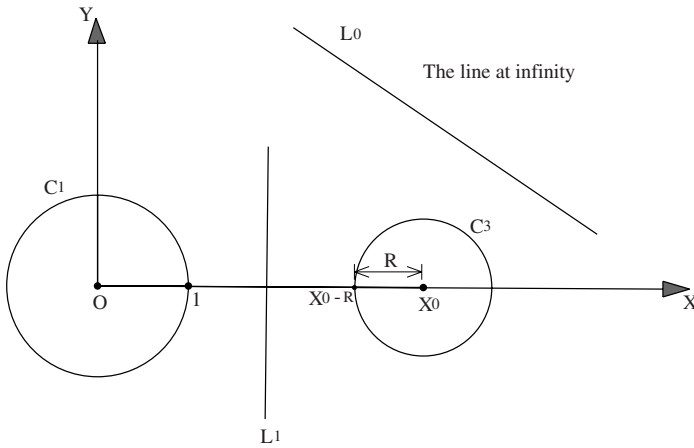


Fig. 3. Two coplanar separate circles \mathbf{C}_1 , \mathbf{C}_3 , and their associated lines \mathbf{L}_1 and \mathbf{L}_0 (the line at infinity). \mathbf{C}_1 and \mathbf{C}_3 intersect at two pairs of conjugate complex points, one pair is on the line \mathbf{L}_1 ; another pair, which is the pair of circular points, is on the line at infinity \mathbf{L}_0 . \mathbf{L}_1 lies between \mathbf{C}_1 and \mathbf{C}_3 , while, \mathbf{L}_0 does not

Theorem 1. *If \mathbf{c}_1 , \mathbf{c}_2 are the images of two parallel circles and separate, their homogeneous equations have two pairs of conjugate complex common solutions. We connect the two points in each pair of the conjugate complex common solutions, then obtain two real lines, called the associated lines of \mathbf{c}_1 and \mathbf{c}_2 . One of the two associated lines lies between \mathbf{c}_1 and \mathbf{c}_2 , and the other one does not lie between \mathbf{c}_1 and \mathbf{c}_2 as shown in Fig. 4. If the camera optical center does not lie between the two parallel planes containing the two circles, the associated line not lying between \mathbf{c}_1 and \mathbf{c}_2 is the vanishing line through the images of circular points. Otherwise, the associated line lying between \mathbf{c}_1 and \mathbf{c}_2 is the vanishing line through the images of circular points.*

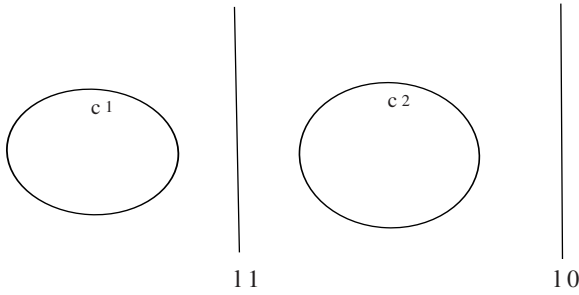


Fig. 4. The distributions of \mathbf{c}_1 , \mathbf{c}_2 , and their two associated lines \mathbf{l}_1 and \mathbf{l}_0 . \mathbf{c}_1 and \mathbf{c}_2 intersect at two pairs of conjugate complex points, one pair is on the line \mathbf{l}_1 ; another pair is on the line \mathbf{l}_0 . \mathbf{l}_1 lies between \mathbf{c}_1 and \mathbf{c}_2 , while, \mathbf{l}_0 does not. The images of circular points are the pair on \mathbf{l}_0 if the optical center does not lie between the two parallel planes containing the two circles. Otherwise, they are the pair on \mathbf{l}_1

Proof. Let the two parallel circles be \mathbf{C}_1 , \mathbf{C}_2 , and P_1 , P_2 be the planes containing them, \mathbf{O} be the camera optical center. Because P_1 , P_2 are parallel, the quadric cone with \mathbf{O} as its vertex and passing through \mathbf{C}_2 intersects the plane P_1 at a circle, denoted by \mathbf{C}_3 . \mathbf{C}_1 and \mathbf{C}_3 are two coplanar circles in P_1 . Because \mathbf{c}_1 and \mathbf{c}_2 are separate, and are also the images of \mathbf{C}_1 and \mathbf{C}_3 , we know that \mathbf{C}_1 and \mathbf{C}_3 are separate too. See Fig. 5. By Lemma 1, there is the fact: one of the associated lines of \mathbf{C}_1 and \mathbf{C}_3 lies between \mathbf{C}_1 and \mathbf{C}_3 (denoted by L), and the other one, i.e. the line at infinity passing through the circular points, does not lie between \mathbf{C}_1 and \mathbf{C}_3 (denoted by L_0).

If \mathbf{O} does not lie between P_1 and P_2 , we know that \mathbf{C}_1 , \mathbf{C}_3 in P_1 are all in front of the camera. Because under a pinhole camera, the imaging process from the parts of P_1 in front of the camera to the image plane is quasi-affine ([7], Chapter 20 in [6]), the relative positions of \mathbf{c}_1 , \mathbf{c}_2 and their associated lines are the same as the ones of \mathbf{C}_1 , \mathbf{C}_3 , L , L_0 . So the associated line not lying between \mathbf{c}_1 and \mathbf{c}_2 is the images of L_0 , i.e. the vanishing line through the images of the circular points.

If \mathbf{O} lies between P_1 and P_2 , the plane through \mathbf{O} and L_0 , denoted by P_0 , which is parallel to P_1 and P_2 , lies between \mathbf{C}_1 and \mathbf{C}_2 . And, the plane through \mathbf{O} and L , denoted by P , does not lie between \mathbf{C}_1 and \mathbf{C}_2 . This is because \mathbf{C}_3 and \mathbf{C}_1 lie on the different sides of P , and also \mathbf{C}_3 and \mathbf{C}_2 lie on the different sides of P . The projection from \mathbf{C}_1 , \mathbf{C}_2 , P_0 , P to their images is quasi-affine, so \mathbf{c}_1 , \mathbf{c}_2 and their associated lines have the same relative positions as the ones of \mathbf{C}_1 , \mathbf{C}_2 , P_0 , P (The image of P_0 is the vanishing line lying between \mathbf{c}_1 and \mathbf{c}_2 , the image of P is the associated line not lying between \mathbf{c}_1 and \mathbf{c}_2).

Then, the theorem is proved.

Therefore, if \mathbf{c}_1 and \mathbf{c}_2 are separate, by Theorem 1, we can find out the images of a pair of circular points. If \mathbf{c}_1 and \mathbf{c}_2 are enclosing, and their two pairs of conjugate complex intersections do not coincide, we can not find out the images of circular points now (if the two pairs of conjugate complex intersections

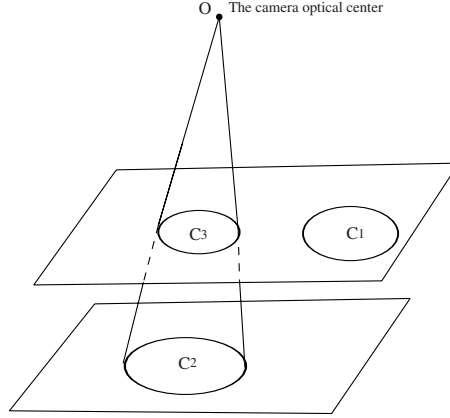


Fig. 5. The camera and two parallel circles C_1 , C_2 . O is the camera optical center. The quadric cone passing through O (as the vertex) and C_2 intersects the plane containing C_1 at another circle C_3 . C_1 and C_3 are coplanar. If the images of C_1 and C_2 are separate, C_1 and C_3 are separate too because the image of C_3 is the same as the image of C_2

coincide to one pair, the coinciding pair is the images of circular points, and at the time, C_1 , C_3 are concentric) [21].

In fact, the enclosing case of c_1 and c_2 usually seldom occurs, and other cases of c_1 and c_2 occur quite often in our life. We regard that c_1 and c_2 are not enclosing below.

By the discussion in the second paragraph in this section and Theorem 1, the images of a pair of circular points can always be determined from a single view of two parallel circles. Assuming the images of the determined circular points to be \mathbf{m}_I , \mathbf{m}_J , by (2), we have two linear equations on the camera intrinsic parameters $\omega = \mathbf{K}^{-\tau} \mathbf{K}^{-1}$ as:

$$\mathbf{m}_I^{\tau} \omega \mathbf{m}_I = 0, \quad \mathbf{m}_J^{\tau} \omega \mathbf{m}_J = 0. \quad (3)$$

If the camera intrinsic parameters are kept unchanged and the motions between cameras are not pure translations, then from three views, six linear equations on the intrinsic parameters can be set up. Thus, the camera can be calibrated completely.

An outline of our algorithm to calibrate a camera from the images of two parallel circles is showed as follows.

- Step 1.* In each view, extract the pixels \mathbf{u} of the images of two parallel circles, and fit them with $\mathbf{u}^{\tau} \mathbf{c}_1 \mathbf{u} = 0$, $\mathbf{u}^{\tau} \mathbf{c}_2 \mathbf{u} = 0$ to obtain \mathbf{c}_1 and \mathbf{c}_2 by the least squares method, then establish two conic equations as $e_1 : \mathbf{x}^{\tau} \mathbf{c}_1 \mathbf{x} = 0$, and $e_2 : \mathbf{x}^{\tau} \mathbf{c}_2 \mathbf{x} = 0$.
- Step 2.* Solve the common solutions of e_1 , e_2 in each view.
- Step 3.* Find out the images of the circular points from the solved common solutions of e_1 , e_2 by the method presented in this section.

Step 4. Set up the equations on the intrinsic parameters $\omega = \mathbf{K}^{-\tau}\mathbf{K}^{-1}$ from the images of circular points found out in Step 3 by (3).

Step 5. Solve out ω from the equations in Step 4 by singular value decomposition method, and then do Cholesky decomposition and inverse the result, or use the equations in [22], to obtain the intrinsic parameters \mathbf{K} .

Remark 1. With the notations \mathbf{O} , L , L_0 as in the proof of Theorem 1, let P be the plane through \mathbf{O} and L , P_0 be the plane through \mathbf{O} and L_0 . By the proof of Theorem 1, we know that the relative positions of \mathbf{C}_1 , \mathbf{C}_2 , P , P_0 are the same as the ones of their images, which just is a **quasi-affine invariance**. For other cases except for the enclosing case, the images of circular points are found out by the real projective invariance preserving the real and conjugate complex intersections of conics respectively. Of course, the projective invariance is also a **quasi-affine invariance**.

Remark 2. In Theorem 1, there are two cases: one case is that the optical center does not lie between the two parallel planes P_1 , P_2 containing the two circles; another case is that the optical center lies between P_1 and P_2 . In general, the former case occurs more often than the latter case. When the two circles are coplanar, the optical center always does not lie between P_1 and P_2 , the associated line of \mathbf{c}_1 and \mathbf{c}_2 not lying between \mathbf{c}_1 and \mathbf{c}_2 is always the vanishing line.

Remark 3. If we use the above method with a calibration grid to calibrate camera in the same way as Zhang's method [24], it might be wise to take two intersecting coplanar circles.

4 Experiments

4.1 Simulated Experiments

In the experiments, the simulated camera has the following intrinsic parameters:

$$\mathbf{K} = \begin{bmatrix} 1500 & 3 & 512 \\ 0 & 1400 & 384 \\ 0 & 0 & 1 \end{bmatrix}.$$

Take two parallel circles in the world coordinates system as: $X^2 + Y^2 = 6^2, Z = 0$; $(X - 20)^2 + Y^2 = 3^2, Z = 10$. And, take three groups of rotation axes, rotation angles and translations as: $\mathbf{r}_1 = (17, 50, 40)^\tau, \theta_1 = 0.3\pi, \mathbf{t}_1 = (-5, 15, 50)^\tau$; $\mathbf{r}_2 = (-50, 50, 160)^\tau, \theta_2 = 0.1\pi, \mathbf{t}_2 = (10, -4, 40)^\tau$; $\mathbf{r}_3 = (90, -70, 20)^\tau, \theta_3 = 0.2\pi, \mathbf{t}_3 = (5, 2, 30)^\tau$. Let \mathbf{R}_i be the rotations from \mathbf{r}_i , θ_i . Then project the two circles to the simulated image planes by the three projective matrices $\mathbf{P}_i = \mathbf{K}[\mathbf{R}_i, \mathbf{t}_i], i = 1, 2, 3$ respectively. The images of the two circles are all separate (in order to verify Theorem 1), and the image sizes are of 700×900 , 550×950 , 500×850 pixels respectively. Gaussian noise with mean 0 and standard deviation ranging from 0 to 2.0 pixels is added to the image points of the two

Table 1. The averages of the estimated intrinsic parameters under different noise levels

Noise levels(pixel)	f_u	f_v	s	u_0	v_0
0	1500.0000	1400.0000	3.0000	511.9999	384.0000
0.4	1500.4522	1400.4622	3.0655	513.0418	384.7355
0.8	1500.3927	1399.8069	2.6278	518.8032	389.0022
1.2	1500.9347	1399.9088	2.8893	525.2412	392.4559
1.6	1501.8536	1399.4255	3.2259	537.6676	399.6847
2.0	1503.9747	1399.7978	2.4428	548.5837	410.4403

Table 2. The RMS errors of the estimated intrinsic parameters under different noise levels

Noise levels(pixel)	f_u	f_v	s	u_0	v_0
0	0.0000	0.0000	0.0000	0.0000	0.0000
0.4	5.1775	4.7679	0.8985	5.1834	5.2046
0.8	11.1786	10.2244	1.8713	11.2057	11.2147
1.2	15.6606	14.3364	2.7034	15.6643	15.9711
1.6	21.1434	19.9497	3.0504	21.4699	21.1630
2.0	26.6784	24.8587	4.8918	27.6128	27.0722

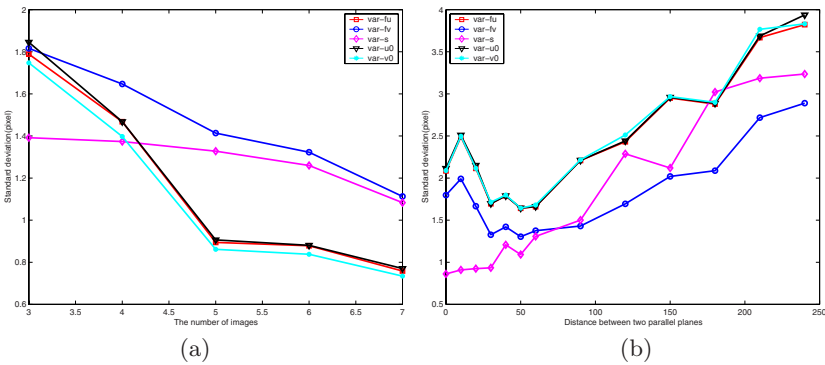


Fig. 6. The standard deviations of the estimated intrinsic parameters vs. (a) the number of images; (b) the distance of the two parallel circles (defined to be the distance of the two parallel planes containing the two circles)

circles, and then the intrinsic parameters are computed. For each noise level, we perform 50 times independent experiments, and the averaged results are shown in Table 1. We also compute the root mean square errors (RMS errors) of intrinsic parameters under different noise levels, the results are given in Table 2.

In order to assess the performance of our calibration technique with the number of images and with the distance of the two parallel circles, the calibrations using 4, 5, 6, 7 images and varying the distance of the two circles are performed respectively, and the standard deviations of the estimated intrinsic parameters are shown in Fig. 6, where the added noise level is 0.5 pixels. It is clear that the deviations tend to decrease with the number of the images increasing. Let d be the distance of the two parallel planes containing the two circles (the two circles used are: $X^2 + Y^2 = 10^2, Z = 0$ and $(X - 40)^2 + Y^2 = 10^2, Z = d$. d is varying from 0 to 240). Then we can see that: (i) the deviations for f_u, f_v, u_0, v_0 tend to decrease with d increasing from 0 to 50, and then to increase with d increasing from 50 to 240; (ii) the deviations for s tend to increase with d increasing. It follows that it is not the coplanar circles such that the algorithm is most stable.



Fig. 7. The used three images of two parallel circles

4.2 Real Experiments

We use a CCD camera to take three photos of two cups as shown in Fig. 7. The photos are of 1024×768 pixels. In each photo, the pixels of the images of the upper circles at the brim of the two cups are extracted, then fitted by the least squares method to obtain two conic equations (see Step 1 of our algorithm). From Fig. 7, we can see that the extracted conics are separate in each view. Applied Theorem 1 and the proposed calibration algorithm to these conic equations, the estimated intrinsic parameter matrix is:

$$\mathbf{K}_1 = \begin{bmatrix} 1409.3835 & 8.0417 & 568.2194 \\ 0 & 1385.3772 & 349.3042 \\ 0 & 0 & 1 \end{bmatrix}.$$

To verify \mathbf{K}_1 , the classical calibration grid DLT method in [1] is used to calibrate the same camera (the intrinsic parameters keep unchanged). The used

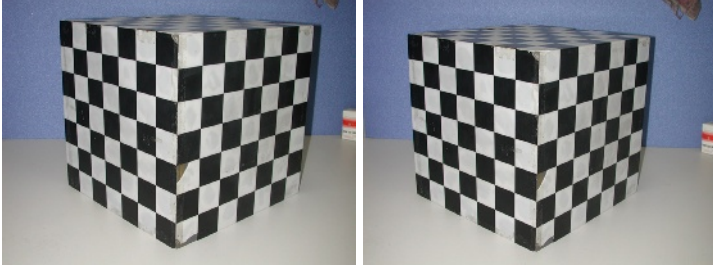


Fig. 8. The used images of a calibration grid

image is the left one in Fig. 8, and the calibration result from 72 corresponding pairs of space and image points is:

$$\mathbf{K}_2 = \begin{bmatrix} 1325.6124 & 4.5399 & 500.7259 \\ 0 & 1321.2270 & 368.4573 \\ 0 & 0 & 1 \end{bmatrix}.$$

The estimated intrinsic parameters \mathbf{K}_1 , \mathbf{K}_2 are used to reconstruct the calibration grid from the two images in Fig. 8. The angles between two reconstructed orthogonal planes are:

$$89.28^\circ \text{ by using } \mathbf{K}_1, \quad 89.97^\circ \text{ by using } \mathbf{K}_2.$$

Both of them are close to the ground truth of 90° . Consider the reconstructed vertical parallel lines on the calibration grid, then compute the angles between any two of them, and the averages are:

$$0.0000476^\circ \text{ by using } \mathbf{K}_1, \quad 0.0000395^\circ \text{ by using } \mathbf{K}_2.$$

Both of them are close to the ground truth of 0° . These results validate the proposed algorithm in this paper.

5 Conclusions

We presented a quasi-affine invariance of two parallel circles, then applied it to calibrating a camera. Both simulated and real experiments were given, and showed the accuracy and robustness of this method. The presented quasi-affine invariance is quite interesting and useful. It can also be applied to recovering the geometry of single axis motions by conic fitting method. We believe that it will have more applications in future.

Acknowledgements. We would like to thank the reviewers for their suggestions. The work is supported by the National Key Basic Research and Development Program (973) under grant No. 2002CB312104 and the National Natural Science Foundation of China under grant No. 60121302.

References

1. Y.I. Abdel-Aziz, and H.M. Karara: Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Proc. ASP/UI Symp. on CloseRange Photogrammetry*, pp. 1–18, 1971.
2. D. Bondyfalat, T. Papadopoulou, and B. Mourrain: Using scene constraints during the calibration procedure. *ICCV*, pp. 124–130, 2001.
3. B. Caprile, and V. Torre: Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2), 127–140, 1990.
4. O.D. Faugeras, Q.T. Luong, and S. Maybank: Camera self-calibration: theory and experiments. *ECCV*, pp. 321–334, 1992.
5. V. Fremont, and R. Chellali: Direct camera calibration using two concentric circles from a single view. *ICAT*, pp. 93–98, 2002.
6. R. Hartley, and A. Zisserman: *Multiple view geometry in computer vision*. Cambridge University press, 2000.
7. R. Hartley: Chirality. *International Journal of Computer Vision*, 26(1), 41–61, 1998.
8. G. Jiang, H.T. Tsui, L. Quan, and A. Zisserman: Single axis geometry by fitting conics. *ECCV*, LNCS 2350, pp. 537–550, 2002.
9. G. Jiang, H.T. Tsui, L. Quan, and S.Q. Liu: Recovering the geometry of single axis motions by conic fitting. *CVPR*, pp. 293–298, 2001.
10. J-S Kim, H-W Kim, and I.S. Kweon: A camera calibration method using concentric circles for vision applications. *ACCV*, pp. 515–520, 2002.
11. D. Liebowitz, and A. Zisserman: Metric rectification for perspective images of planes. *CVPR*, pp. 482–488, 1998.
12. D. Liebowitz, and A. Zisserman: Combining scene and auto-calibration constraints. *ICCV*, pp. 293–300, 1999.
13. Q.T. Luong, and O.D. Faugeras: Self-calibration of a moving camera from point correspondence and fundamental matrices. *International Journal of Computer Vision*, 22(3), 261–289, 1997.
14. X.Q. Meng, and Z.Y. Hu: A new easy camera calibration technique based on circular points. *Pattern Recognition*, 36(5), 1155–1164, 2003.
15. M. Pollefeys, R. Koch, and L. Van Gool: Self-calibration and metric reconstruction in spite of varying and unknown intrinsic camera parameters. *ICCV*, pp. 90–95, 1998.
16. L. Quan: Invariant of a pair of non-coplanar conics in space: definition, geometric interpretation and computation. *ICCV*, pp. 926–931, 1995.
17. J.G. Semple, and G.T. Kneebone: *Algebraic projective geometry*. Oxford University Press, 1952.
18. P. Sturm, and S. Maybank: On plane-based camera calibration: a general algorithm, singularities, applications. *CVPR*, pp. 432–437, 1999.
19. R.Y. Tsai: A versatile camera calibration technique for accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4), 323–344, 1987.
20. M. Wilczkowiak, E. Boyer, and P. Sturm: Camera calibration and 3D reconstruction from single images using parallelepipeds. *ICCV*, pp. 142–148, 2001.
21. Y.H. Wu, X.J. Li, F.C. Wu, and Z.Y. Hu: Coplanar circles, quasi-affine invariance and calibration. <http://www.nlpr.ia.ac.cn/english/rv/~yhwu/papers.htm>, 2002.
22. Y.H. Wu, and Z.Y. Hu: A new constraint on the imaged absolute conic from aspect ratio. <http://www.nlpr.ia.ac.cn/english/rv/~yhwu/papers.htm>, 2003.

23. C.J. Yang, F.M. Sun, and Z.Y. Hu: Planar conic based camera calibration. *ICPR*, pp. 555–558, 2000.
24. Z. Zhang: A flexible new technique for camera calibration. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(11), 1330–1334, 2000.
25. Z. Zhang: Camera calibration with one-dimensional objects. *ECCV*, pp. 161–174, 2002.

Appendix: Proof of Lemma 1

\mathbf{C}_1 , \mathbf{C}_3 are coplanar and separate, then we can set up the Euclidean coordinate system as: one of the centers of \mathbf{C}_1 and \mathbf{C}_3 as the origin \mathbf{O} , the line through the two centers as the X -axis, the line through \mathbf{O} and orthogonal to the X -axis as the Y -axis, the radius of one of the circles as the unit length. For example, we take the coordinate system as in Fig. 3. Then the homogeneous equations of \mathbf{C}_1 and \mathbf{C}_3 are respectively:

$$X^2 + Y^2 = Z^2, \quad (X - X_0Z)^2 + Y^2 = R^2Z^2 \quad (4)$$

where R is the radius of \mathbf{C}_3 , X_0 horizontal coordinate of the center of \mathbf{C}_3 . Because \mathbf{C}_1 and \mathbf{C}_3 separate, $X_0 > 1 + R$. Solve the common solutions of (4), and compute the associated lines, then we have them as:

$$\mathbf{L}_1 : X = \frac{X_0^2 - R^2 + 1}{2X_0}, \quad \mathbf{L}_0 : Z = 0 \quad (\text{the line at infinity})$$

Because $X_0 > 1 + R$, we can prove that the following inequality holds:

$$1 < \frac{X_0^2 - R^2 + 1}{2X_0} < X_0 - R < \infty$$

From the inequality, we know that \mathbf{C}_1 , \mathbf{C}_3 lie on the different sides of \mathbf{L}_1 . Since \mathbf{L}_0 is at infinity, \mathbf{C}_1 , \mathbf{C}_3 must lie on the same side of it as shown in Fig. 3.

In addition, the above proof is independent of the chosen Euclidean coordinate system. This is because if we set up another Euclidean coordinate system (with the same or different unit length as the above one), they can be transformed each other by a Euclidean transformation, or a similarity transformation, which preserves the line at infinity and preserves the relative positions of objects.

Texton Correlation for Recognition

Thomas Leung

Fujifilm Software

1740 Technology Drive, Suite 490, San Jose, CA 95110, U.S.A.

tleung@fujifilmsoft.com

Abstract. We study the problem of object, in particular face, recognition under varying imaging conditions. Objects are represented using local characteristic features called textons. Appearance variations due to changing conditions are encoded by the correlations between the textons. We propose two solutions to model these correlations. The first one assumes locational independence. We call it the conditional texton distribution model. The second captures the second order variations across locations using Fisher linear discriminant analysis. We call it the Fisher texton model. Our two models are effective in the problem of face recognition from a single image across a wide range of illuminations, poses, and time.

1 Introduction

Recognition under varying imaging conditions is a very important, yet challenging problem. Imaging conditions can change due to external and internal factors. External factors include illumination conditions (back-lit vs. front-lit or overcast vs. direct sunlight) and camera poses (frontal view vs. side view). Internal variations can arise from time (natural material weathers or rusts, or people aging) or internal states (facial expressions or a landscape changing appearance according to the season). The changes an object exhibits under varying imaging conditions are usually referred to as within-class variations in pattern recognition.

The ability to be invariant to within-class variations determines how successful an algorithm will be in practical applications. In recent years, a lot of attention in the research community has been devoted to this problem. Some representative examples and their application domains are (1) generic 3-D objects [11]; (2) faces [1,2,3,6,7]; and (3) natural materials [4,10,14].

In this paper, we strive to develop algorithms to recognize objects, in particular faces, under varying imaging conditions. The fundamental observation comes from human vision. After seeing many objects under different conditions, humans build an implicit internal model of how objects change their appearance. Using this model, humans can *hallucinate* any object's appearance under novel conditions. For example, one can easily recognize a person from the side after seeing only a single frontal picture of this person. Or, one can still recognize a friend with ease after not seeing him for 10 years. Of course, recognition is not always perfect, especially under some unusual conditions, but the accuracy is significant.

We adopt a learning framework to build a model of how the appearance of objects change under different imaging conditions. We call it the texton correlation model.

Textons are a discrete set of representative local features for the objects. The basic idea is to encode efficiently how the textons transform when illumination, camera pose, etc... change. Taking into account these transformations, we can build a similarity measure between images which are insensitive to imaging conditions. Using the texton correlation model, our algorithms can recognize faces from a single image of a person under a wide range of illuminations and poses, and also after many years of aging.

The outline of this paper is as follows. The concept of textons is reviewed in Section 2. Assuming locational independence, we propose a solution to capture the within-class variations using the *conditional texton distribution*. Experimental results using the conditional texton distribution are presented in Section 4. In Section 5, we introduce the idea of *Fisher textons* to capture second-order correlations across both pixel locations and imaging conditions. Results using *Fisher textons* are also presented. Finally, we conclude and discuss future work in Section 6.

2 Textons

Julesz [9] first proposed to use the term texton to describe the putative units of preattentive human texture perception. Julesz's textons — orientation elements, crossings, and terminators — lack a precise definition for gray level images. Recently, the concept of textons has been re-invented and operationalized. Leung and Malik [10] define textons as learned co-occurrences of outputs of linear oriented Gaussian derivative filters. Variations of this concept have been applied to the problem of 3D texture recognition [4,10,14]. We adopt a similar definition of textons in this paper.

What textons encode is a discrete set of local characteristic features of a 3D surface in the image space. This discrete set is referred to as the vocabulary. Every location on the image is mapped to an element in this vocabulary. For example, if the 3D surface is a human face, one texton may encode the appearance of an eye, another the mouth corner. For natural materials such as concrete, the textons may encode the image characteristic of a bar, a ridge, or a shadow edge. The textons can be learned from a single class (e.g. John, or concrete), thus forming a class-specific vocabulary. It can also be learned from a collection of classes (e.g. {John, Mary, Peter, ...}, or {concrete, velvet, plaster, ...}), thus forming a universal vocabulary. One advantage of the discrete nature of the texton representation is the ability to characterize changes in the image due to variations in imaging conditions easily. For example, when a person changes from a smiling expression to a frown, the mouth corner may change from texton element I to element J . Or when the illumination moves from a frontal direction to an oblique angle, the element on a concrete surface may transform from texton element A to element B . The main focus of this paper is to study how to represent this texton element transformation to recognize objects and materials under varying imaging conditions.

In this paper, textons are computed in the following manner. First, the image is filtered with a filterbank of linear Gaussian derivative filters. Specifically, the filters are the horizontal and vertical derivatives of circular symmetric Gaussian filters:

$$F_V(\sigma) = \frac{d}{dx} G_\sigma(x, y)$$

$$F_H(\sigma) = \frac{d}{dy} G_\sigma(x, y)$$

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma} \exp\left(-\frac{x^2 + y^2}{2\sigma}\right)$$

4 different scales are used, giving a total of 8 filters. The particular choice of filters is not very important¹. This set is selected for their simplicity and ease of computation. In fact, the filters are x-y separable and filtering can be done in $O(N)$ time instead of $O(N^2)$ time, where N is the dimension of the kernel. After filtering, each pixel is transformed into a vector of filter responses of length 8. These filter responses are clustered using the K-means algorithm [5] to produce K prototypical features for the objects. With this vocabulary, every pixel in an image is mapped to the closest texton element according to the Euclidean distance in the filter output space. We call the output of this process the texton labels for an image. The value at each pixel is now between 1 and K , depending on which texton best describes the local surface characteristics.

3 Conditional Texton Distribution

We represent the texton transformation in a probabilistic formulation. The objective here is to capture how objects, faces, or natural materials change their appearance under varying imaging conditions. The goal is to learn the intrinsic transformation which is valid for all the instances within an object class. For example, in the context of face recognition, we would learn the transformation from a training set of a large group of people. The intrinsic variations within a single person and the differences between individuals are captured in the model. This learned transformation can be applied to any group of novel subjects and recognition can be achieved from a single image.

Let M be the image of a model. For example, in face recognition, M will be an image of the person you want to recognize. Let I be an incoming image. The task is to determine whether I is the same object as M . Let T_M be the texton labels for M and T_I be that of I . We define $P_{same}(T_I|T_M)$ to be the probability that I is the same object as the model M . Similarly, we define $P_{diff}(T_I|T_M)$ to be the probability that it is a different object. The likelihood ratio can be used to determine whether they come from the same object:

$$L(T_I|T_M) = \frac{P_{same}(T_I|T_M)}{P_{diff}(T_I|T_M)} \quad (1)$$

The task is to define $P_{same}(T_I|T_M)$ and $P_{diff}(T_I|T_M)$. We make the simple assumption that the texton labels are independent of their location:

$$P_{same}(T_I|T_M) = \prod_x P_{same}(T_I(x)|T_M(x))$$

$$P_{diff}(T_I|T_M) = \prod_x P_{diff}(T_I(x)|T_M(x))$$

¹ Other filter choices can be found in [10,13,14]

The likelihood ratio can be used as a similarity measure between an image and the model. We can either set a threshold on $L(T_I|T_M)$ to determine whether the face matches the model, or as in classification, assign the incoming image to the class with the highest likelihood ratio score, L .

3.1 Learning the Distribution from Data

The discrete nature of textons allows us to represent the distributions $P(T_I(x)|T_M(x))$ exactly, without making simplifying assumptions such as a Gaussian distribution. Notice that $T_I(x)$ is an element of the texton vocabulary and is scalar-valued: $T_I(x) \in \{1, \dots, K\}$. In fact, with a texton vocabulary of size K , $P(T_I(x)|T_M(x))$ can be represented completely as an $K \times K$ table. This conditional probability table can be easily learned through training data.

Let the training set be \mathcal{T} . Let \mathcal{C}_M be the set of all training data that belong to the same class as M . Let $a, b \in \{1, \dots, K\}$ be two texton elements in the vocabulary. The entries in the probability table can be accumulated as follows²:

$$P_{same}(T_I = a|T_M = b) = \frac{1}{Z_1} \sum_{M, I \in \mathcal{T}} \mathbf{1}_{(a,b,\mathcal{C}_M)}(T_I, T_M, I)$$

$$P_{diff}(T_I = a|T_M = b) = \frac{1}{Z_2} \sum_{M, I \in \mathcal{T}} \bar{\mathbf{1}}_{(a,b,\mathcal{C}_M)}(T_I, T_M, I)$$

where Z_1 and Z_2 are normalizing constants to make P_{same} and P_{diff} probabilities. The function $\mathbf{1}_{(a,b,\mathcal{C}_M)}(T_I, T_M, C_I) = 1$ if $T_I = a, T_M = b, I \in \mathcal{C}_M$ and 0 otherwise. $\bar{\mathbf{1}}_{(a,b,\mathcal{C}_M)} = 1$ if $T_I = a, T_M = b, I \notin \mathcal{C}_M$ and 0 otherwise.

Applying these two learned conditional probability tables to the likelihood ratio $L(T_I|T_M)$ in Eq. 1, the similarity between a model and an incoming image can be computed.

4 Experiments

In this section, we will describe results of applying the conditional texton distribution model to the problem of face recognition. Before images are compared, faces are automatically extracted by a face detector [8]. Eyes and mouth corners are found using a similar algorithm to normalize each face for size and in-plane rotation. Each face is resampled to a standard size of 30×30 pixels. In all the experiments in this paper, separate texton vocabularies are learned independently at each pixel location. The main reason for this choice is the speed needed to compute texton assignment. For each pixel, a vocabulary size of 10 is used. In total, there are $30 \times 30 \times 10 = 9000$ textons altogether.

In all the experiments, the training set is used to obtain the texton vocabularies and the conditional texton distributions. All results are reported on a disjoint test set, in which none of the individuals appear in the training set. Results will be reported on two applications: face verification and face classification. First, we describe the databases used in this paper.

² The x dependency is implied implicitly to make the notations more readable.

4.1 Database

There are 3 face databases used to evaluate the performance under different imaging condition:

Yale face database: Using a geodesic dome with 64 flashes, researchers at Yale University collected a database of objects and human faces [2]. The set of face images is used in this paper to evaluate the performance of our algorithm towards varying illuminations. There are 15 individuals. Images in which the illumination angle is within $\pm 60^\circ$ in elevation and azimuth are used. These 32 images are shown in Figure 1. This database will be referred to as *yale* in this paper.



Fig. 1. The Yale face database. 32 illumination directions are used. The top two rows correspond to illumination directions randomly chosen to form the training set. The bottom two rows correspond to those in the test set.

FERET: The FERET training database [12] consists of faces of 1203 individuals, each with several frontal images. For a subset of the individuals (670 people), non-frontal images are available. We break up this database into two sets to measure the performance of our algorithm separately. The *Frontal FERET* consists of faces with different expressions and slightly different natural lighting. Robustness of our algorithm towards out-of-plane rotations (up to $\pm 45^\circ$) will be measured using the *Rotated FERET* subset.

ID photos: This database consists of employee ID photos at a Japanese company. There are 836 individuals, with a total of 1834 images. All the employees are Asians, with the majority Japanese. Every individual has at least 2 images, one taken at the time of hire and another taken recently³. For each individual, there is a large age difference in the different photographs — usually several years, up to as many as 20 years. This is a challenging database because people can change their appearance significantly over the span of several years: from wearing glasses to wearing contact lenses, from skinny to fleshy, from having a lot of hair to bald, etc. This database will be able to test how our algorithm performs when people change their appearance when aging. Since these are ID photographs, lighting is well-controlled, though not constant

³ Some have one more photo taken during their employment.

across all photographs. Expression is usually neutral. This database will be referred to as *ID* in this paper. Due to privacy issues, only photos of 2 individuals can be shown in Figure 2.



Fig. 2. Examples of *ID* photos. Two different photographs of the same individual can be taken up to 20 years apart. This database can test our algorithm's performance against large age differences.

4.2 Face Verification

We first consider the problem of face verification. One application of face verification is an access control security system. A user inputs his/her identity through an ID card. A camera will capture an image of the user. The face will be detected and matched against the one previously stored in the system. If the match is good, the user will be granted access, otherwise denied. The basic concept is to compare two faces and decide whether it is the same person. In all our experiments, we randomly select two faces from the test set. These two faces can come from two different individuals or from the same individual⁴. Accuracy is measured by the false positive and false negative rates, in the form of a ROC (receiver operating characteristics) curve. All experiments are repeated using random partitions of the databases into training and test sets. Results reported are the average performance.

We first study the performance of our algorithm with respect to illumination variations using the *yale* database. 10 subjects are used for training and the remaining 5 subjects for testing. 16 illumination directions are chosen randomly and the corresponding images are used as training. These illumination directions are the same for every training individual. The corresponding images for one person are shown in the top two rows in Figure 1. The bottom two rows show the illumination directions for the test set. There is a complete disjoint between the training set and the test set. In other words, *none* of the illumination directions in the test set is present in the training set for any individual. Our algorithm performs perfectly in this experiment, getting 0% false positive and 0% false negative. Notice that the two images to be verified can have light directions up to 120° apart. This means the texton distribution does a very good job encoding the intrinsic changes in feature appearance under different illuminations. The learned distribution extrapolates well for both new individuals and novel illumination directions.

⁴ For the case of the same individual, the two images are never identical.

The performance of our algorithm on the *ID*, *Frontal FERET*, and *Rotated FERET* databases is shown in Figure 3. Figure 3(b) is the zoomed-in version of (a). The blue, red, and green curves are the ROC curves for the *Frontal FERET*, *Rotated FERET*, and *ID* databases respectively. The equal error rate (false positive rate equals to false negative rate) for the three experiments are 4.2%, 10.5%, and 9% respectively.

The dashed black curve in Figure 3 is the ROC curve on the *Frontal FERET* database with a system trained using *ID* pictures. The idea is to see how well the learned texton distributions generalize to a different database. Most machine learning algorithms guarantee good generalization only if the statistics of the training and test sets is identical. In this case, the statistics can be quite different, for example, the ethnic composition of the subjects are totally different: *ID* contains pictures of predominantly Japanese, while *Frontal FERET* contains pictures with diverse ethnic backgrounds.

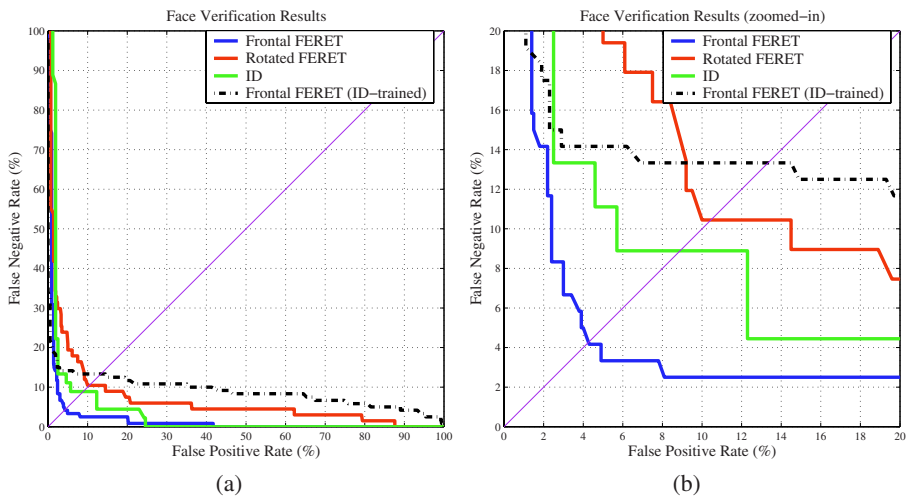


Fig. 3. Face verification results. The blue, red, and green curves are the ROC curves for the *Frontal FERET*, *Rotated FERET*, and *ID* databases respectively. For each of these three experiments, the training and test sets come from the same database. The dashed black curve indicates the ROC curve for the *Frontal FERET* database using a system trained with the *ID* photos.

4.3 Face Classification

In this section, we investigate the effectiveness of our conditional texton distribution model for the problem of face classification. Let there be P individuals, each with a single image as the model. For any new image of these P people, we want to automatically determine who he is. We will use the similarity measure in Section 3 (Equation 1) and classify this new image into the model with the highest similarity score.

For all the databases, we randomly pick P individuals from the test set. For each person, one image is randomly selected to be the model, another to be the probe. The

model and the probe can be of vastly different imaging conditions. This procedure is repeated multiple times for different choices of P individuals and different random training and test sets. Our algorithm works perfectly for the *yale* database, giving 0% error. The results for the other databases are reported in Figure 4. The blue, red, and green curves report the error rates for the *Frontal FERET*, *Rotated FERET*, and *ID* databases respectively. In these three cases, the training and test sets come from the same database. We would like to emphasize that for all the experiments, only a single image is used for the model. This is a very difficult problem, especially for the *Rotated FERET* case, because the probe can be up to 90° out-of-plane rotated from the model. The black curve in Figure 4 presents the error rate on *Frontal FERET* with a system trained using the *ID* photos. This indicates the effectiveness of our algorithm when the statistics of the test set (predominantly Japanese) is different from that of the training set (ethnically diverse).

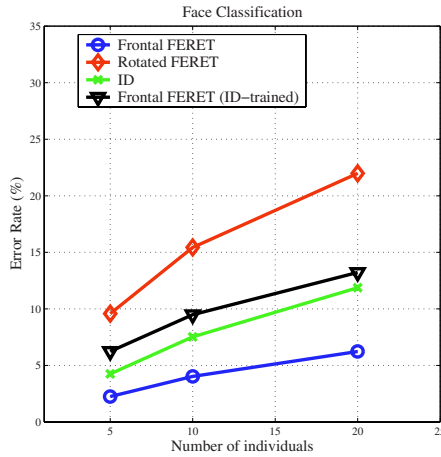


Fig. 4. Face classification results. The model consists of one image. The blue, red, and green curves represent the error rates for the *Frontal FERET*, *Rotated FERET*, and *ID* databases respectively. The black curve represents the error rate for the *Frontal FERET* with a system trained using *ID* photos.

5 Fisher Textons

The conditional texton distribution model presented in Section 3 makes the assumption that the texton assignments are independent of location. This is obviously a wrong assumption. For example, the appearance of the left eye and the right eye are definitely correlated. However, this assumption enables us to compute the likelihood ratio (Equation 1) efficiently.

In this section, we explore the correlation between locations on the face. Specifically, we take into account second order correlations. After texton assignment, every face is

turned into a 30×30 array of texton labels. Each pixel, $T_I(x)$, takes on the value between $1, \dots, K$, where K is the size of the texton vocabulary at each pixel⁵. We transform each pixel to an indicator vector of length K : $[0, \dots, 0, 1, 0, \dots, 0]$ with 1 at the k^{th} element if $T_I(x) = k$. We concatenate all the vectors together, so that each image becomes a $30 \times 30 \times 10 = 9000$ dimensional vector.

We perform the Fisher linear discriminant analysis [5] on these vectors to obtain the projection directions which are best for separating faces from different people. Specifically, let the within-class scatter matrix be:

$$\begin{aligned} S_w &= \sum_{i=1}^c S_i \\ S_i &= \sum_{\mathbf{x} \in \mathcal{C}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t \\ \mathbf{m}_i &= \frac{1}{n_i} \sum_{\mathbf{x} \in \mathcal{C}_i} \mathbf{x} \end{aligned}$$

where c is the number of classes and \mathcal{C}_i is the set of training examples belonging to class i . $n_i = |\mathcal{C}_i|$ and $n = \sum_i^c n_i$. The between-class scatter matrix is defined as:

$$S_b = \sum_{i=1}^c n_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^t$$

where \mathbf{m} is the total mean vector:

$$\mathbf{m} = \frac{1}{n} \sum_{i=1}^c n_i \mathbf{m}_i$$

The objective is to find \mathbf{V} to maximize the following criterion function:

$$J(\mathbf{V}) = \frac{|\mathbf{V}^t \mathbf{S}_b \mathbf{V}|}{|\mathbf{V}^t \mathbf{S}_w \mathbf{V}|}$$

The columns of the optimal \mathbf{V} are the generalized eigenvectors corresponding to the largest eigenvalues in

$$\mathbf{S}_b \mathbf{v}_i = \lambda_i \mathbf{S}_w \mathbf{v}_i$$

The vectors \mathbf{v}_i are the projection directions which capture the essential information to classify objects among different classes. The idea is that when a large number of training examples are used, the \mathbf{v}_i 's can distinguish between people not only those present in the training set.

We call these projection vectors \mathbf{v}_i the *Fisher textons*. Every incoming image is transformed into an N dimensional vector by projecting into these Fisher textons. Similarity between two faces is taken simply as the Euclidean distance between the projections.

⁵ $K = 10$ in this paper.

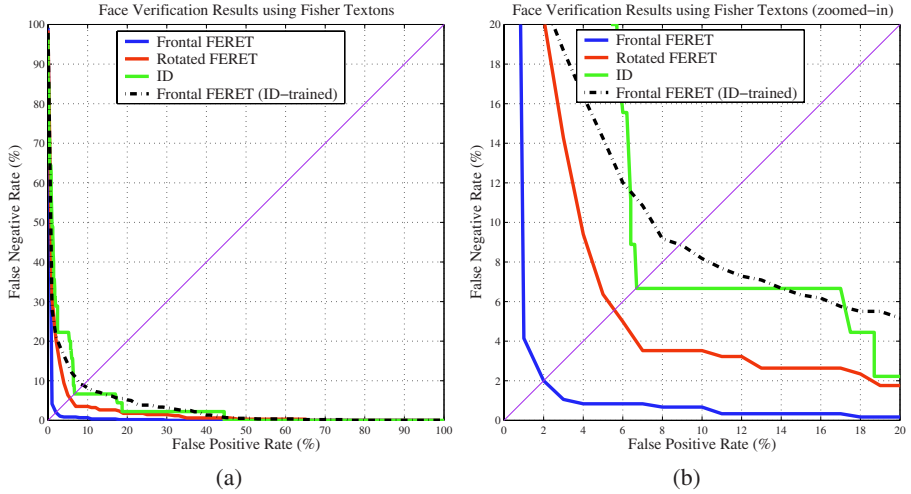


Fig. 5. Face verification results using Fisher textons. The blue, red, and green curves are the ROC curves for the *Frontal FERET*, *Rotated FERET*, and *ID* databases respectively. For each of these three experiments, the training and test sets come from the same database. The dashed black curve indicates the ROC curve for the *Frontal FERET* database using a system trained with the *ID* photos.

Let us contrast the differences between the Fisher textons and the conditional texton distribution. For Fisher textons, locational correlations are captured up to second order. However, imaging condition correlations are captured only up to the second order as well. On the other hand, the texton conditional distribution model sacrifices location dependencies to capture the exact texton distributions under changing imaging conditions.

The results on the problem of face verification are shown in Figures 5. The blue, red, and green curves are the ROC curves for the *Frontal FERET*, *Rotated FERET*, and *ID* databases respectively. The equal error rates are 2%, 5.5%, 6.6% respectively. The dashed black curve indicates the ROC curve for the *Frontal FERET* database using a system trained with the *ID* photos, with an equal error rate of 9%. The performance on the face verification task is uniformly better than that produced by the conditional texton distribution model. The added locational dependencies more than offset the sacrifice made on the imaging condition correlations.

Results for face classification using Fisher textons are shown in Figure 6. The performance is better for the *Frontal FERET* (blue curve) and *Rotated FERET* (red curve) databases. However, it is worse for *ID* (green curve) and *Frontal FERET* database when trained using *ID* photos (black curve). One possible explanation is that the within-class variations in the *ID* database is so large (because of the long timeline) that just capturing the second order correlations is not enough to distinguish individuals well. Using the whole distribution, as in the case of the conditional texton distribution model, can thus produce better results.

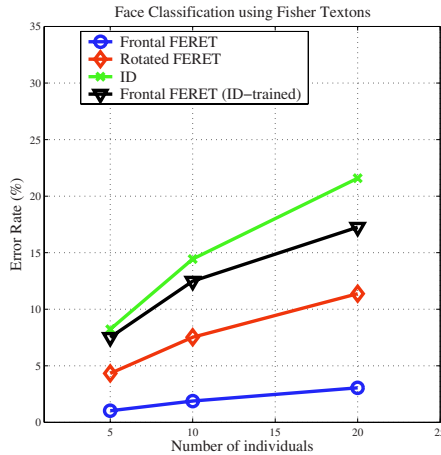


Fig. 6. Face classification results using Fisher Textons. The blue, red, and green curves represent the error rates for the *Frontal FERET*, *Rotated FERET*, and *ID* databases respectively. The black curve represents the error rate for the *Frontal FERET* with a system trained using *ID* photos.

6 Conclusions

In this paper, we study the problem of recognition under varying imaging conditions. Two algorithms are proposed based on the idea of textons. The first algorithm is to use conditional texton distributions to model the within-class and between-class variations exactly. But, the assumption of locational independence is made. We call the second algorithm Fisher textons. Second order correlations in both location and imaging condition variations are captured. Both algorithms are effective in the problems of face verification and recognition.

Comparing with state-of-the-art algorithms is difficult without training and testing on the same datasets. Future work includes thorough comparisons with other algorithms. Another direction for future work is to develop algorithms to capture the exact dependencies from both pixel locations and changing imaging conditions. The obvious choice is a Markov Random Field. However, the parameters in MRFs are difficult to estimate without a large quantity of data. Inference is not a trivial task either. Finding efficient ways to capture the correlations will be an interesting problem from both theoretical and practical points of view.

Acknowledgements. Portions of the research in this paper use face images collected at Yale University and under the FERET program. The author would like to thank David Forsyth, Jitendra Malik, Sergey Ioffe, Troy Chinen, Yang Song, and Ken Brown for helpful discussions.

References

1. V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(9), 2003.
2. H. Chen, P. Belhumeur, and D. Jacobs. In search of illumination invariants. In *Proc. Conf. CVPR*, 2000.
3. T. Cootes, K. Walker, and C.J. Taylor. View-based active appearance models. In *Proc. Intl. Conf. Automatic Face and Gesture Recognition*, 2000.
4. O. Cula and K. Dana. Compact representation of bidirectional texture functions. In *Proc. Computer Vision and Pattern Recognition*, pages 1041–7, 2001.
5. R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, 2001.
6. A. Georgiades, P. Belhumeur, and D. Kriegman. From few to many: illumination cone models for face recognition under varying lighting and pose. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(6), 2001.
7. R. Gross, L. Matthews, and S. Baker. Eigen light-fields and face recognition across pose. In *Proc. Int'l. Conf. Automatic Face and Gesture Recognition*, 2002.
8. S. Ioffe. Automatic red-eye reduction. In *Proc. Int. Conf. Image Processing*, 2003.
9. B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–7, March 1981.
10. T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. J. Computer Vision*, 43(1):29–44, 2001.
11. H. Murase and S. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal on Computer Vision*, 14(1):5–24, 1995.
12. P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.
13. C. Schmid. Constructing models for content-based image retrieval. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2001.
14. M. Varma and A. Zisserman. Classifying images of materials: achieving viewpoint and illumination independence. In *Proc. European Conference Computer Vision*, pages 255–71, 2002.

Multiple View Feature Descriptors from Image Sequences via Kernel Principal Component Analysis

Jason Meltzer¹, Ming-Hsuan Yang², Rakesh Gupta², and Stefano Soatto¹

¹ University of California, Los Angeles, CA 90095, USA,

² Honda Research Institute, Mountain View, CA 94041, USA

Abstract. We present a method for learning feature descriptors using multiple images, motivated by the problems of mobile robot navigation and localization. The technique uses the relative simplicity of small baseline tracking in image sequences to develop descriptors suitable for the more challenging task of wide baseline matching across significant viewpoint changes. The variations in the appearance of each feature are learned using kernel principal component analysis (KPCA) over the course of image sequences. An approximate version of KPCA is applied to reduce the computational complexity of the algorithms and yield a compact representation. Our experiments demonstrate robustness to wide appearance variations on non-planar surfaces, including changes in illumination, viewpoint, scale, and geometry of the scene.

1 Introduction

Many computer vision problems involve the determination and correspondence of distinctive regions of interest in images. In the area of robot navigation, a mobile platform can move through its environment while observing the world with a video camera. In order to determine its location, it must create a model that is rich enough to capture this information yet sparse enough to be stored and computed efficiently. By dealing with only sparse image statistics, called *features*, these algorithms can be made more efficient and robust to a number of environmental variations that might otherwise be confusing, such as lighting and occlusions. Usually, these features must be tracked across many images to integrate geometric information in space and time. Thus, one must be able to find correspondences among sets of features, leading to the idea of *descriptors* which provide distinctive signatures of distinct locations in space. By finding features and their associated descriptors, the correspondence problem can be addressed (or at least made simpler) by comparing feature descriptors.

In applications such as N-view stereo or recognition, it is frequently the case that a sparse set of widely separated views are presented as input. For such *wide baseline* problems, it is necessary to develop descriptors that can be derived from a single view, since no assumptions can be made about the relative viewpoints among images. In contrast, in the cases of robot navigation or real-time structure

from motion, a video stream is available, making it possible to exploit *small baseline* correspondence by tracking feature locations between closely spaced images. This provides the opportunity to incorporate multiple views of a feature into its signature, since that information is present and easily obtained. Under these circumstances, the problem of tracking across frames is relatively simple, since the inter-frame motion is small and appearance changes are minimal. Many existing feature trackers, such as [12,35,37], can produce chains of correspondences by incrementally following small baseline changes between images in the sequence. These trackers, however, are far from ideal and often drift significantly within a handful frames of motion, thus they cannot be used for wide baseline matching. In order to reduce this effect, features must be recognized despite a variety of appearance changes. Additionally, when a sudden large change in viewpoint occurs, it is necessary to relate features that have already been observed to their counterparts in a new image, thus maintaining the consistency of the model that the features support. In this paper, we propose to integrate the techniques of short baseline tracking and wide baseline matching into a unified framework which can be applied to correspondence problems when video streams are available as input.

One may question the necessity of such a multi-view descriptor, given that many single view features work well (see the next section for an overview). In some cases, in fact, multiple views will add little to the descriptor matching process, since the variability can be well modelled by a translational or affine transformation. When these assumptions are violated, such as when there are non-planar surfaces in a scene, or complex 3D geometry, multiple views of a feature can render significant robustness to viewpoint changes. When the geometry itself is changing, such a descriptor is necessary to capture this variability. A multiple view descriptor can provide a generic viewpoint, meaning the results of matching will be less sensitive to particular viewpoints (which may be confusing special cases). Perhaps the most compelling argument in favor of the multi-view approach is that, in many applications, the data is already there. When this is the case, it makes sense to try to leverage the data available, rather than discard it after processing each frame.

1.1 Related Work

The problem of finding and representing distinctive image features for the purposes of tracking, reconstruction, and recognition is a long-standing one. Recently, a number of authors (Schmid *et al* [7,23], Lowe [21,20], Baumberg [1], Tuytelaars and Van Gool [38,39], Schaffalitzky and Zisserman [28,29]) have developed *affine invariant* descriptors of image locations. These expand upon the pioneering work of Lindeberg [19], Koenderink [15], and others who study image scale space and its properties. The general approach of these methods is to find image locations which can be reliably detected by searching for extrema in the scale space of the image [19]. Given different images of a scene taken over small or wide baselines, such descriptors can be extracted independently on each pair, then compared to find local point correspondences.

Lowe's scale invariant feature transform (SIFT) [20] considers an isotropic Gaussian scale space, searching for extrema over scale in a one-dimensional scale space. The difference-of-Gaussian function is convolved with the image, computed by taking the difference of adjacent Gaussian-smoothed images in the scale space pyramid. Once the extrema are located (by finding maxima and minima in neighborhoods of the scale space pyramid), they are filtered for stability then assigned a canonical orientation, scale and a descriptor derived from gradient orientations in a local window. The descriptor is adapted from Edelman *et al* [8], which models the outputs of "complex cells" that respond to particular image gradients within their receptive fields. In a similar manner, SIFT samples gradients around the points of interest and amalgamates them into a 4x4 array of 8-bin orientation histograms. This constitutes the 128 element SIFT descriptor. SIFT has been shown to match reliably across a wide range of scales and orientation changes, as well as limited 3D perspective variation [24].

Mikolajczyk and Schmid's affine invariant interest point detector [23] seeks to find stable areas of interest in the affine image scale space, which has three parameters of scale. It first selects initial feature locations using a multi-scale Harris corner detector [12] then applies an iterative technique to find the best location, scale, and shape transformation of the feature neighborhood. The procedure converges to yield a point location in the image, as well as a canonical transformation which can be used to match the feature despite arbitrary affine transformations of its neighborhood. Descriptors consist of normalized Gaussian derivatives computed on patches around the points of interest. These patches are transformed by the canonical mapping used in the detection phase of the algorithm, which yields scale and skew invariance. Rotational invariance is obtained using steerable filters, and affine photometric invariance is achieved by normalizing all of the derivatives by the first. A dimension 12 descriptor is the final output of the procedure, involving derivatives up to 4th order.

Tuytelaars and Van Gool [39] have developed methods which explicitly take into account a variety of image elements, such as corners, edges, and intensity. They find and characterize affinely invariant neighborhoods by exploiting properties of these elements, then match similar points using geometric and photometric constraints to prune false matches. An explicit assumption they make is that the areas of interest selected lie on approximately planar regions, though their experiments demonstrate robustness to violations of this assumption.

The aforementioned works (which represent only a small fraction of the latest literature on the topic, see also [1,7,9,5,12,14,15,20,21,22,23,26,27,28,29,34,36,38,39,40] for some others) focus on the problem of extracting properties from local patches in a *single* image in order to match these locations in subsequent images of the same scene. In [9], Ferrari *et al* present a method for matching features across multiple unordered views, which is derived from pairwise view matches using the detector described in [39]. When a video stream is available, as often is the case when using cameras on mobile robots, more information is present than can be obtained from a single image of a scene or an arbitrary set of such images. It is therefore reasonable to seek a *multi-view descriptor*, which incorporates information from across multiple adjacent views of a scene to yield better feature correspondences.

1.2 Rationale and Overview of the Approach

We address the wide-baseline correspondence problem under the specific scenario of autonomous guidance and navigation, where high frame-rate video is available during both training (“map building”) and localization, but viewing conditions can change significantly between the two. Such changes affect both the domain of the image (geometric distortion due to changes of the viewpoint and possibly deformations of the scene) and its range (changes in illumination, deviation from Lambertian reflection). If $w : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ denotes a piecewise smooth function, and $I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^+$ denotes the image, then in general two given images are related by $I_2(x) = \rho(I_1(w(x)))$ where ρ is a functional that describes range deformations, and w describes domain deformations. Such changes are due to both intrinsic properties of the scene ξ (shape, reflectance) and to *nuisance factors* ν (illumination, viewpoint), so we write formally $\rho = \rho_{\xi, \nu}(I)$ and $w = w_{\xi, \nu}(x)$. The goal of the correspondence process is to associate different images to a common cause (the same underlying scene ξ) despite the nuisances ν .

A “feature” is a statistic of the image, $\phi : I \rightarrow \mathbb{R}^k$ that is designed to facilitate the correspondence process.¹ Ideally one would want a feature statistic that is invariant with respect to all the nuisance factors: $\phi \circ \rho_{\xi, \nu}(I(w_{\xi, \nu}(x))) = f_{\xi}(x)$ independent of ν , for some function f and for all allowable nuisances ν . Unfortunately this is not possible in general, since there exists no single-view statistic that is invariant with respect to viewpoint or lighting conditions, even for Lambertian scenes. Nuisances that can be moded-out in the representation are called *invertible*.² What nuisance is invertible depends on the representation of the data. If we consider the data to be a single image, for instance, viewpoint is not an invertible nuisance. However, if multiple adjacent views of the same scene are available, as for instance in a video from a moving camera, then viewpoint can be explicitly accounted for, at least in theory. Additionally, changes in viewpoint elicit irradiance changes that are due to the interplay of reflectance and illumination, and it is therefore possible that “insensitive” (if not invariant) features can be constructed. This is our rationale for designing feature descriptors that are based *not* on single views, but on multiple adjacent views of the same scene.

Any correspondence process relies on an underlying model of the scene, whether this is stated explicitly or not: our model of the scene is a constellation of planar patches that support a radiance density which obeys a diffuse+specular reflection model. This means that for patches that are small enough one either sees the Lambertian albedo, or a reflection of the light source, and therefore the rank of an aggregation of corresponding views is limited [13]. We represent

¹ Facilitation should be quantified in terms of computational complexity, since the benefit of using features to establish correspondence is undermined by Rao-Blackwell’s theorem, that guarantees that a decision based on any statistic of the data achieves a conditional risk that is no less than the decision based on the raw data.

² Euclidean, affine, and projective image motion are invertible nuisances, in the sense that they can be eliminated by pre-processing the image. However, viewpoint is not invertible, unless the scene has special symmetries that are known *a priori*.

the multi-view descriptor by a rank-constraint in a suitable inner product of a deformed version of the image: $\text{rank}(\mathcal{T}) = r$, where the tensor \mathcal{T} is defined by

$$\mathcal{T} \doteq (\Phi(I_1(w_1(x))), \Phi(I_2(w_2(x))), \dots, \Phi(I_n(w_n(x)))) \quad (1)$$

for a suitable function Φ that maps the image to a higher-dimensional space.³ The modeling “responsibility” is shared by the transformation w and the map Φ : the more elaborate the one, the simpler the other. What is not modeled explicitly by w and Φ goes to increase the rank of \mathcal{T} . In this general modeling philosophy, our approach is therefore broadly related to [6,10,11]. In this work, we consider the following combinations:

Translational domain deformation, generic kernel: We use the simplest possible $w(x) = x + T$, a generic map Φ , and all the responsibility for modeling deformations of the domain and the range is relegated to the principal components of the tensor \mathcal{T} . In this case, the descriptor is invariant with respect to plane-translations, but all other deformations contribute to the rank $r \simeq n$.

Affine domain deformation, generic kernel: $w(x) = Ax + T$, and additional geometric distortion due to changes of viewpoint and photometric variability is relegated to the principal components of \mathcal{T} .

Viewpoint deformation: In this case, $w(x)$ depends on the 3-D structure of the scene, and is explicitly inverted by projective reconstruction and normalization. Therefore, \mathcal{T} is viewpoint invariant and its principal components only models photometric variability.

2 Proposed Solution

We relegate deformations in the images not accounted for by the transformation w to the principal components of \mathcal{T} . Principal component analysis (PCA) operates on the premise that a low dimensional basis suffices to approximate the covariance matrix of the samples, thereby providing a compact representation. Given M observation images, PCA diagonalizes the covariance matrix $C = \frac{1}{M} \sum_{j=1}^M \mathbf{y}_j \mathbf{y}_j^T$ (where \mathbf{y}_j can be considered a vectorized image patch, and without loss of generality we assume that \mathbf{y} is pre-processed to be zero mean) by solving an eigenvalue equation [2]. The Karhunen-Loeve (KL) transform is an efficient method to compute the basis (principal components), which can be carried out using singular value decomposition (SVD) [3]. For the case where we have a stream of incoming data, the sequential Karhunen-Loeve algorithm exploits the low dimension approximation characteristic by partitioning and transforming the data into blocks of orthonormal columns to reduce computational and memory

³ We use Φ to indicate the map from the data to what is known in the kernel-machine community as “feature space” and ϕ for the feature that we have defined in this section (i.e. an invariant image statistic). The notation should not be confusing in the end, since ϕ will comprise principal components of $\Phi(I_j)$.

requirements [18,4]. In other words, this algorithm essentially avoids the computation of a full-scale SVD at every time step by only applying the necessary computation to smaller data blocks for updating the KL basis incrementally.

2.1 Incremental Update of Kernel Principal Components

Since PCA aims to find an optimal low dimensional linear subspace that minimizes the reconstruction error, it does not perform well if the data points are generated from a nonlinear manifold. Furthermore, PCA encodes the data based on second order dependencies (pixel-wise covariance among the pixels), and ignores higher-order statistics including nonlinear relations among the pixel intensity values, such as the relationships among three or more pixels in an edge or a curve, which can capture important information for recognition.

The shortcomings of PCA can be overcome through the use of kernel functions with a method called kernel principal component analysis (KPCA). In contrast to conventional PCA which operates in the input image space, KPCA performs the same procedure as PCA in a high dimensional space, F , related to the input by the (nonlinear) map $\Phi : \mathbb{R}^N \rightarrow F$, $\mathbf{y} \mapsto Y$.⁴ If one considers $\mathbf{y} \in \mathbb{R}^N$ to be a (vectorized) image patch, $Y \in F$ is this image patch mapped into F . The covariance matrix for M vectors in F is $C' = \frac{1}{M} \sum_{j=1}^M \Phi(\mathbf{y}_j)\Phi(\mathbf{y}_j)^T$, assuming $\sum_{k=1}^M \Phi(\mathbf{y}_k) = 0$ (see [31] for a method to center $\Phi(\mathbf{y})$). By diagonalizing C' , a basis of kernel principal components (KPCs) is found. As demonstrated in [30], by using an appropriate kernel function $k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$, one can avoid computing the inner product in the high-dimensional space F . The KPCs are implicitly represented in terms of the inputs (image patches) \mathbf{y} , the kernel k , and a set of linear coefficients β , as $\Psi = \sum_{i=1}^M \beta_i \Phi(\mathbf{y}_i)$, $\Psi \in F$.

To choose an appropriate kernel function, one can either estimate it from data or select it *a priori*. In this work, we chose the Gaussian kernel

$$k(\mathbf{w}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{w} - \mathbf{y}\|^2}{2\sigma^2}\right) \quad (2)$$

based on empirical study, though a principled yet computationally expensive method for learning the kernel from data using quadratic programming was recently demonstrated by Lanckriet et al. [17].

Unfortunately, there is no “online” version of KPCA, as exists for standard PCA [4,18]. In order to avoid computations in the high-dimensional space, all computations are performed through the kernel in terms of linear combinations of input vectors. Hence, in traditional KPCA, all of the input vectors (image patches) must be stored in order to perform classification. This is unacceptable for feature descriptors, since the storage requirement is high, the computational complexity grows with more examples, and the representation is not compact.

⁴ F is typically referred to as “feature space,” but to avoid confusion we will refrain from using that name.

To avoid this problem, we have employed an online (i.e. constant time and memory) version, *approximate* KPCA, which continually produces a fixed number of approximations of the input patches. These approximations and their expansion coefficients, along with the kernel function, form a compact representation of the KPCA basis in high-dimensional space, which can be used to compare an observed feature with this descriptor. The technique is based on the one of Schölkopf *et al* [33] to find approximate “pre-images” of vectors implicitly represented in high-dimensional space. Given a vector $\Psi = \sum_{i=1}^M \alpha_i \Phi(\mathbf{y}_i)$, we seek an approximation $\Psi^* = \sum_{i=1}^L \beta_i \Phi(\mathbf{z}_i)$ with $L < M$. For a particular pre-image \mathbf{z} , [33] demonstrates that it is sufficient to minimize the distance between Ψ and its projection onto $\Phi(\mathbf{z})$, which is equivalent to maximizing

$$\frac{\langle \Psi, \Phi(\mathbf{z}) \rangle^2}{\langle \Phi(\mathbf{z}), \Phi(\mathbf{z}) \rangle}. \quad (3)$$

By differentiating and substituting the Gaussian kernel function for inner products, the following fixed-point iteration for \mathbf{z} is obtained:

$$\mathbf{z}^{n+1} = \frac{\langle \mathbf{Y}, (\alpha \cdot * \mathbf{K}^n) \rangle}{\alpha^T \mathbf{K}^n} \quad (4)$$

where \mathbf{Y} is a matrix of input vectors \mathbf{y}_i (image patches), α is the vector of coefficients, \mathbf{K}^n is the vector of kernels $[k(\mathbf{y}_1, \mathbf{z}^n), \dots, k(\mathbf{y}_N, \mathbf{z}^n)]^T$, and $\cdot *$ is the element-wise multiplication operator. In order to find a number of such approximations, $\mathbf{z}_1, \dots, \mathbf{z}_M$, we set $\Psi^{m+1} = \Psi^m - \beta^m \Phi(\mathbf{z}^m)$, where \mathbf{z}^m is found using (4). One can solve for the optimal coefficients of the expansion, β , in each iteration to yield $\Psi^* = \sum_{i=1}^L \beta_i \Phi(\mathbf{z}_i)$ [32].

In order to match a newly observed image to existing descriptors, our algorithm searches the image for patches which have a small residual when projected onto the stored KPCA descriptors. That is, it finds \mathbf{y} , a patch from the new image, and ψ , a descriptor (KPCA basis), such that the following is minimized for a choice of ψ and \mathbf{y} .

$$\left\| \Phi(\mathbf{y}) - \sum_{i=1}^N \frac{\langle \Phi(\mathbf{y}), \psi_i \rangle}{\langle \psi_i, \psi_i \rangle} \psi_i \right\|^2 \quad (5)$$

where ψ_i is a kernel principal component and N is the number of components in the descriptor.

2.2 Feature Descriptors through Incremental Update of KPCA

Our method for extracting feature descriptors from image sequences proceeds as follows:

1. **Bootstrap with a small-baseline tracker:** Read a number of frames of the input sequence, track the features using a standard tracking method, and store the image patches of each feature. As a translation-invariant tracker, $w(x) = x + T$, we use Lukas and Kanade’s [37] classic algorithm; for affine-invariant tracker, $w(x) = Ax + T$, we use the Shi and Tomasi (ST) algorithm.

2. **Construct kernel basis:** Perform KPCA using the Gaussian kernel separately on each feature’s training sequence or reduced training set found in step 3.
3. **Approximate kernel basis:** Form an approximate basis for each feature by finding approximate patches which lead to the least residual estimate of the original basis in high-dimensional space. Create L such patches for each feature. In our algorithm, L is a tuning parameter. Further discussion of “pre-image” approximation in KPCA can be found in [32,33].

The above algorithm yields a set of descriptors, each corresponding to a particular feature. Given a novel image of the same scene, these descriptors can be matched to corresponding locations on the new image using (5).

3 Experiments

We performed a variety of experiments with the proposed system to test the efficacy of small and wide baseline correspondence. In the experiments, two phases were established: *training* and *matching*, which correspond to short and wide baseline correspondence. In the training phase, a video sequence was recorded of a non-planar object; this object underwent a combination of 3D rotation, scaling, and warping with respect to the camera. The Shi-Tomasi (ST) [35] tracker (or Lucas-Kanade (LK), in the case of translational tracking) was used to obtain an initial set of points, then the procedure of the previous section was used to track these locations and develop feature descriptors via approximate KPCA. Note that we do not show experiments for projective reconstruction, since we did not see any benefit to this approach using our data sets. In future experiments with more severe depth variations, we expect to see significant benefits by normalizing the projective reconstruction.

In the matching phase, a test image from outside the training sequence was used to find wide-baseline correspondences. First, initial features were selected using the ST or LK selection mechanism. The purpose of this was to find the most promising locations based on the same criteria used in the tracking phase. In the case of affine tracking, the ST tracking algorithm then performed affine warping in a neighborhood around each candidate point, seeking to minimize the discrepancy between this point and each stored in the training phase. In the translational case, no such warping was applied. The quality of a candidate match was calculated by finding the projection distance of this patch onto the basis of the descriptor using (5). Finally, candidate matches that fell below a threshold distance were selected, and the best among those was chosen as the matching location on the test image.

The results for matching are displayed in the figures using an image of the training sequence for reference, but the matching process does not use that image. Rather, it matches the descriptors derived from all of the video frames. Any image in the training sequence could be used for this visualization.

It is important to note a few aspects of our experiments. First, we are only concerned with feature *selection* or *tracking* insofar as they influence the ex-

perimental quality. Any consistent feature selector and tracker can be used to estimate the candidate points, or even no feature tracker at all (the entire image or a neighborhood could be searched, for example). For computational reasons, we used the ST selector and tracker for the experiments, which proved robust enough for our purposes.

A number of tuning parameters are present in the algorithms. For the LK and ST trackers, one must choose a threshold of point selection, which bounds the minimum of the eigenvalues of the second moment matrix of the image around the point. This was set, along with a minimum spacing between points of 10 pixels, such that about 50 points were selected on the object of interest. During tracking, a pyramid of scales is used to impose scale invariance. We chose to calculate three levels of this pyramid in all experiments, hence the pyramid included the image and downsampled versions 1/2 and 1/4 of its original size. For the KPCA algorithm, the tuning parameters are S , the size of the image patches, N , the number of kernel principal components to keep, L , the number of approximate patches to keep (to support the approximate kernel principal component basis), and σ , the bandwidth of the Gaussian kernel. We used $S = 31 \times 31$, $N = 8$ for the translational case, $N = 3$ for the affine case, $L = 4$, and $\sigma = 70$. These were selected by experiment.

While we did not optimize our experiments for speed or programming efficiency, we found the average time for tracking between adjacent frames to be approximately 10 seconds. The code was executed in Matlab on a 1.7GHz Pentium IV processor. The video frames and test images were 640x480 8-bit greyscale pixels, and about 50 feature locations were tracked every frame. The wide baseline matching had similar time requirements, taking up to twenty minutes to match 50 features, since a brute force combinatorial search was used. Optimizations in compiled code, as well as search heuristics, would increase these speeds dramatically. We have developed a C++ version of the tracking phase of this code, which runs at speeds of greater than 15Hz on the same hardware.

The figures in the following pages show a selection of many experiments. In all figures, lines link corresponding points between views. The results were pruned of matches outside the relevant objects to make the figures less cluttered. When comparing the choice of tracker, we found that affine tracking required fewer principal components than translational tracking to produce similar correspondence rates. When attempting to match scenes that are rotated or scaled with respect to the training sequence, the affine tracking scheme has a clear advantage, since these domain transformation are explicitly accounted for by the tracking mechanism. Such variability could not be represented in the translational case unless it was observed in the training sequence.

4 Concluding Remarks

We have presented a novel method for extracting feature descriptors from image sequences and matching these to new views of a scene. Rather than derive invariance completely from a model, our system learns the variability in the images

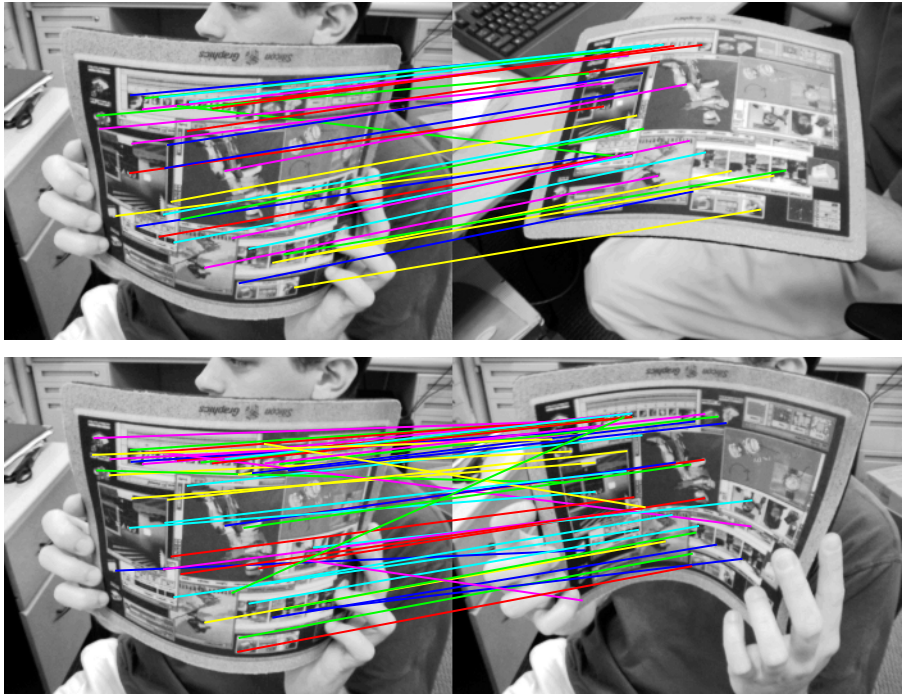


Fig. 1. Affine tracking + KPCA: A non-planar surface undergoing warping and viewpoint changes. (Top-left) The first image of the training sequence. (Top-right) The test image, outside of the training sequence. 27 feature locations were correctly matched, with 2 false positives. (Bottom-left) Image from the training sequence. (Bottom-right) The warped object. 40 locations correctly matched, 6 false positives.

directly from data. This technique is applicable in situations where such data is already available, such as robot navigation, causal structure from motion, face tracking, and object recognition.

There are a number of ways in which our system can be extended. Because a kernel technique is used, we must approximate the basis comprising the feature descriptor with virtual inputs. The best way to do this remains an open problem, and we are investigating other methods in addition to the one presented here ([16, 25], for example). When tracking and matching, we use the ST selector to provide an initial guess for feature locations. While convenient, this may not be the best choice, and we are experimenting with the use of more modern feature selectors ([20,23,28]). In cases where the observed scene is known to be rigid, robust structure from motion techniques (RANSAC or a robust Kalman Filter, for example) can be used to remove incorrect correspondences and suggest potential feature locations. Finally, the experimental code must be translated into more efficient form, allowing it to be used in near real-time on mobile platforms.

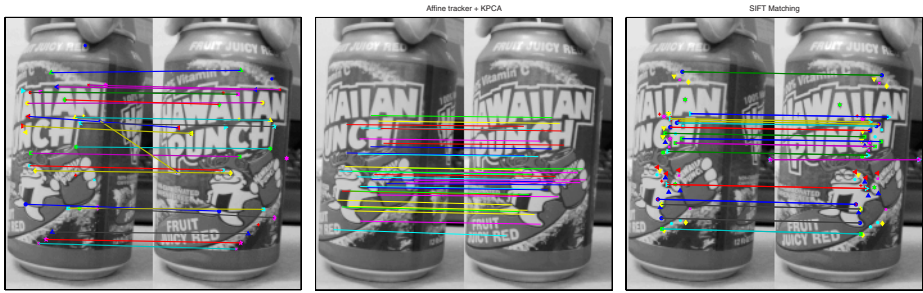


Fig. 2. Translational, Affine, and SIFT: The figures show the results of matching using our algorithms and SIFT on a rotating can. (Left) The translational Multi-View technique correctly matches a number of features near the edges of the can. There are 22 correct correspondences and 8 false positives. (Center) The affine Multi-View matches 32 locations and zero outliers, but with fewer matches along the edges. (Right) SIFT correctly matches more locations overall, but toward the center portion of the can where the transformation is approximately affine. Some lines between matches are included for clarity.



Fig. 3. Translational + KPCA: Matching using a non-planar surface undergoing warping and scale changes. During training, the object was moved away from the camera and deformed. (Left) shows the first image of the training sequence. (Right) shows the test image, which is outside of the training sequence. The shapes and colors indicate the corresponding points. In this example, 50 feature locations were correctly matched, with 10 false positives.

Acknowledgements. This work was conducted while the first author was an intern at the Honda Research Institute in 2003. Meltzer and Soatto are supported in part by the following grants: AFOSR F49620-03-1-0095, ONR N00014-03-1-0850, NSF ECS-0200511, and NSF CCR-0121778. The authors thank David Ross, David Lowe, and the Intel OpenCV project for providing code.

References

1. A. Baumberg. "Reliable feature matching across widely separated views," *Proc. CVPR*, 2000.
2. P. Belhumeur, J. Hespanha, D. Kriegman. "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *Proc. ECCV*, 1996.
3. C. Bishop. *Neural Networks for Pattern Recognition*. Oxford U. Press, 1995.
4. J. Bunch and C. Nielsen. "Updating the singular value decomposition." *Numerische Mathematik*, 31:111–129, 1978.
5. G. Carneiro and A. Jepson. "Phase-based local features." *Proc ECCV*, 2002.
6. T. Cootes, G. Wheeler, K. Walker and C. Taylor. "View-Based Active Appearance Models." *Image and Vision Computing*, Vol.20, 2002, pp. 657–664.
7. Y. Dufournaud, C. Schmid, R. Horaud. "Matching images with different resolutions," *Proc. CVPR*, 2000.
8. S. Edelman, N. Intrator, and T. Poggio. "Complex cells and object recognition," unpublished manuscript, 1997.
9. V. Ferrari, T. Tuytelaars and L. Van Gool. "Wide-baseline multiple-view correspondences." *Proc. CVPR*, Madison, USA, June 2003.
10. A. Fitzgibbon and A. Zisserman. "Joint manifold distance: a new approach to appearance based clustering." *Proc. CVPR*, 2003.
11. B. Frey and N. Jojic. "Transformed Component Analysis: Joint Estimation of Spatial Transformations and Image Components." *Proc. ICCV*, 1999.
12. C. Harris and M. Stephens. "A combined corner and edge detector." *Alvey Vision Conference*, 1988.
13. H. Jin, S. Soatto, A. Yezzi. "Multi-view stereo beyond Lambert." *CVPR*, 2003.
14. A. Johnson and M. Herbert. "Object recognition by matching oriented points." *CVPR*, 1997.
15. J. Koenderink and A. van Doorn. "Generic neighborhood operators," *IEEE Trans. Pattern Analysis and Machine Intell.*, vol. 14, pp. 597–605, June 1992.
16. J. Kwok and I. Tsang. "The Pre-Image Problem in Kernel Methods." *Proc. 20th Int. Conf. on Machine Learning*, 2003.
17. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, M. Jordan. "Learning the kernel matrix with semi-definite programming." *Proc. 19th Int. Conf. on Machine Learning*, Sydney, Australia, 2002.
18. A. Levy, M. Lindenbaum. "Sequential Karhunen-Loeve Basis Extraction and its Application to Images." *IEEE Trans. on Image Processing*, Aug. 2000.
19. T. Lindeberg. *Scale-Space Theory in Computer Vision*. Boston: Kluwer Academic Publishers, 1994.
20. D. Lowe. "Distinctive image features from scale-invariant keypoints," Preprint, submitted to *IJCV*. Version date: June 2003.
21. D. Lowe. "Object recognition from local scale-invariant features," *Proc. ICCV*, Corfu, Greece, September 1999.
22. K. Mikolajczyk and C. Schmid. "Indexing based on scale invariant interest points." *Proc. 8th ICCV*, 2001.
23. K. Mikolajczyk, C. Schmid. "An affine invariant interest point detector." *Proc. ECCV*, 2002.
24. K. Mikolajczyk and C. Schmid. "A performance evaluation of local descriptors." *Proc. CVPR*, June 2003.
25. D. Pavlov, D. Chudova, and P. Smyth. "Towards scalable support vector machines using squashing." *Proc. Int. Conf. on Knowledge Discovery in Databases*, 2000.

26. P. Pritchett and A. Zisserman. "Wide baseline stereo matching." *6th ICCV*, 1998.
27. F. Rothganger *et al.* "3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints." *Proc. CVPR*, June 2003.
28. F. Schaffalitzky and A. Zisserman. "Viewpoint invariant texture matching and wide baseline stereo," *Proc. ICCV*, Jul 2001.
29. F. Schaffalitzky and A. Zisserman. "Multi-view matching for unordered image sets, or 'How do I organize my holiday snaps?'" *7th ECCV*, Copenhagen, 2002.
30. B. Schölkopf, A. Smola. *Learning with Kernels*. Cambridge: The MIT Press, 2002.
31. B. Schölkopf, A. Smola, K. Müller. "Nonlinear component analysis as a kernel eigenvalue problem." *Neural Computation*, 10, 1299–1319, 1998.
32. B. Schölkopf *et al.* "Input Space vs. Feature Space in Kernel-Based Methods," *IEEE Transactions on Neural Networks*. 1999.
33. B. Schölkopf, P. Knirsch, A. Smola and C. Burges, "Fast Approximation of Support Vector Kernel Expansions, and an Interpretation of Clustering as Approximation in Feature Spaces", *DAGM Symposium Mustererkennung*, Springer Lecture Notes in Computer Science, 1998.
34. C. Schmid and R. Mohr. "Local Greyvalue Invariants for Image Retrieval." *Pattern Analysis and Machine Intelligence*, 1997.
35. J. Shi and C. Tomasi. "Good Features to Track," *IEEE CVPR*, 1994.
36. D. Tell and S. Carlsson. "Wide Baseline Point Matching Using Affine Invariants Computed from Intensity Profiles." *Proc. 6th ECCV*, 2000.
37. C. Tomasi, T. Kanade. "Detection and tracking of point features." Tech. Rept. CMU-CS-91132. Pittsburgh: Carnegie Mellon U. School of Computer Science, 1991.
38. T. Tuytelaars and L. Van Gool. "Wide Baseline Stereo based on Local, Affinely invariant Regions," *British Machine Vision Conference*, pp. 412–422, 2000.
39. T. Tuytelaars, and L. Van Gool. "Matching Widely Separated Views based on Affine Invariant Regions," to appear in *Int. J. on Computer Vision*, 2004.
40. R. Zabih and J. Woodfill. "Non-Parametric Local Transforms for Computing Visual Correspondence." *Proc. ECCV*, 1994.

An Affine Invariant Salient Region Detector

Timor Kadir, Andrew Zisserman, and Michael Brady

Department of Engineering Science,
University of Oxford,
Oxford, UK.
`{timork,az,jmb}@robots.ox.ac.uk`

Abstract. In this paper we describe a novel technique for detecting salient regions in an image. The detector is a generalization to affine invariance of the method introduced by Kadir and Brady [10]. The detector deems a region salient if it exhibits unpredictability in both its attributes and its spatial scale.

The detector has significantly different properties to operators based on kernel convolution, and we examine three aspects of its behaviour: invariance to viewpoint change; insensitivity to image perturbations; and repeatability under intra-class variation. Previous work has, on the whole, concentrated on viewpoint invariance. A second contribution of this paper is to propose a performance test for evaluating the two other aspects. We compare the performance of the saliency detector to other standard detectors including an affine invariance interest point detector. It is demonstrated that the saliency detector has comparable viewpoint invariance performance, but superior insensitivity to perturbations and intra-class variation performance for images of certain object classes.

1 Introduction

The selection of a set of image regions forms the first step in many computer vision algorithms, for example for computing image correspondences [2,17,19,20,22], or for learning object categories [1,3,4,23]. Two key issues face the algorithm designer: the subset of the image selected for subsequent analysis and the representation of the subset. In this paper we concentrate on the first of these issues. The optimal choice for region selection depends on the application. However, there are three broad classes of image change under which good performance may be required:

1. Global transformations. Features should be repeatable across the expected class of global image transformations. These include both geometric and photometric transformations that arise due to changes in the imaging conditions. For example, region detection should be covariant with viewpoint as illustrated in Figure 1. In short, we require the segmentation to commute with viewpoint change.

2. Local perturbations. Features should be insensitive to classes of semi-local image disturbances. For example, a feature responding to the eye of a human face should be unaffected by any motion of the mouth. A second class of disturbance is

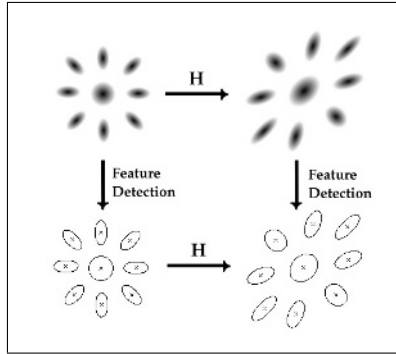


Fig. 1. Detected regions, illustrated by a centre point and boundary, should commute with viewpoint change – here represented by the transformation H .

where a region neighbours a foreground/background boundary. The detector can be required to detect the foreground region despite changes in the background.

3. Intra-class variations. Features should capture corresponding object parts under intra-class variations in objects. For example, the headlight of a car for different brands of car (imaged from the same viewpoint).

In this paper we make two contributions. First, in Section 2 we describe extensions to the region detector developed by Kadir and Brady [10]. The extensions include covariance to affine transformations (the first of the requirements above), and an improved implementation which takes account of anti-aliasing. The performance of the affine covariant region detector is assessed in Section 3 on standard test images, and compared to other state of the art detectors.

The second contribution is in specifying a performance measure for the two other requirements above, namely tolerance to local image perturbations and to intra-class variation. This measure is described in Section 4 and, again, performance is compared against other standard region operators.

Previous methods of region detection have largely concentrated on the first requirement. So-called corner features or interest points have had wide application for matching and recognition [7,21]. Recently, inspired by the pioneering work of Lindeberg [14], scale and affine adapted versions have been developed [2,18,19,20]. Such methods have proved to be robust to significant variations in viewpoint. However, they operate with relatively large support regions and are potentially susceptible to semi-local variations in the image; for example, movements of objects in a scene. They fail on criterion 2.

Moreover, such methods adopt a relatively narrow definition of saliency and scale; scale is usually defined with respect to a convolution kernel (typically a Gaussian) and saliency to an extremum in filter response. While it is certainly the case that there are many useful image features that can be defined in such a manner, efforts to generalise such methods to capture a broader range of salient image regions have had limited success.

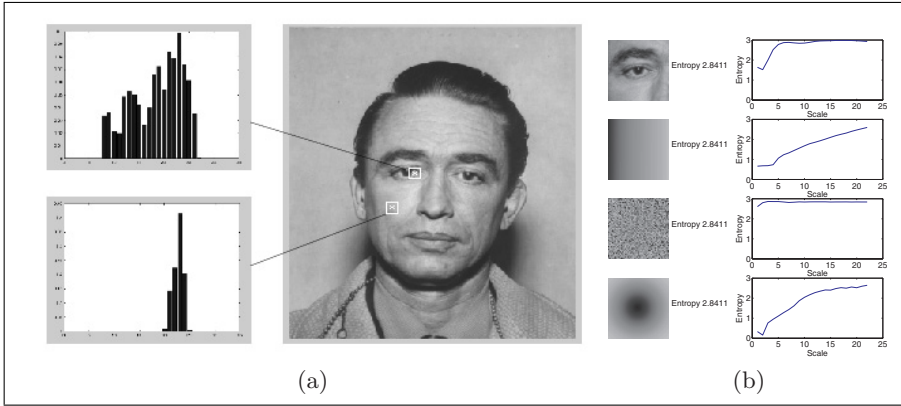


Fig. 2. (a) Complex regions, such as the eye, exhibit unpredictable local intensity hence high entropy. Image from NIST Special Database 18, Mugshot Identification Database. However, entropy is invariant to permutations of the local patch (b).

Other methods have extracted affine covariant regions by analysing the image isocontours directly [17,22] in a manner akin to watershed segmentation. Related methods have been used previously to extract features from mammograms [13]. Such methods have the advantage that they do not rely on excessive smoothing of the image and hence capture precise object boundaries. Scale here is defined in terms of the image isocontours rather than with respect to a convolution kernel or sampling window.

2 Information Theoretic Saliency

In this section we describe the saliency region detector. First, we review the approach of Kadir and Brady [10], then in Section 2.2 we extend the method to be affine invariant, and give implementation details in Sections 2.3 and 2.4.

2.1 Similarity Invariant Saliency

The key principle underlying the Kadir and Brady approach [10] is that salient image regions exhibit unpredictability, or ‘surprise’, in their local attributes *and* over spatial scale. The method consists of three steps: I. Calculation of Shannon entropy of local image attributes (e.g. intensity or colour) over a range of scales — $\mathcal{H}_D(s)$; II. Select scales at which the entropy over scale function exhibits a peak — \mathbf{s}_p ; III. Calculate the magnitude change of the PDF as a function of scale at each peak — $\mathcal{W}_D(s)$. The final saliency is the product of $\mathcal{H}_D(s)$ and $\mathcal{W}_D(s)$ at each peak. The histogram of pixel values within a circular window of radius s , is used as an estimate of the local PDF. Steps I and III measure the feature-space and the inter-scale predictability respectively, while step II selects optimal scales. We discuss each of these steps next.

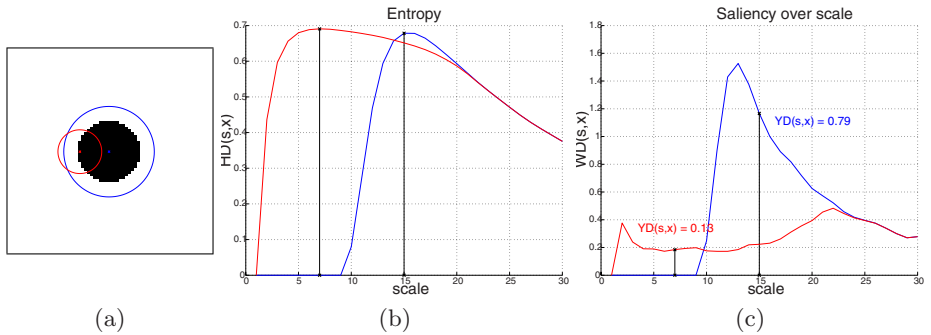


Fig. 3. The two entropy peaks shown in (a) correspond to the centre (in blue) and edge (in red) points in top image. Both peaks occur at similar magnitudes.

The entropy of local attributes measures the predictability of a region with respect to an assumed model of simplicity. In the case of entropy of pixel intensities, the model of simplicity corresponds to a piecewise constant region. For example, in Figure 2(a), at the particular scales shown, the PDF of intensities in the cheek region is peaked. This indicates that most of these pixels are highly predictable, hence entropy is low. However, the PDF in the eye region is flatter which indicates that here, pixel values are highly unpredictable and this corresponds to high entropy.

In step II, scales are selected at which the entropy is peaked. Through searching for such extrema, the feature-space saliency is locally optimised. Moreover, since entropy is maximised when the PDF is flat, i.e. all present attribute values are in equal proportion, such peaks typically occur at scales where the statistics of two (or more) different pixel populations contribute equally to the PDF estimate. Figure 3(b) shows entropy as a function of scale for two points in Figure 3(a). The peaks in entropy occur at scales for which there are equal proportions of black and white pixels present. These significant, or salient scales, in the entropy function (analogous to the ‘critical-points’ in Gaussian scale-space [11,15]) serve as useful reference points since they are covariant with isotropic scaling, invariant to rotation and translation, and robust to small affine shears.

Note however, that the peaks for both points in Figure 3(b) attain an almost identical magnitude. This is to be expected since both patches contain almost identical proportions of black and white pixels. In fact, since histogramming destroys all local ordering information all permutations of the local patch do not affect its entropy. Figure 2(b) shows the entropy over scale function for an image patch taken from 2(a) and three permutations of its pixels: a linear ramp, a random reordering and a radial gradient. The entropy at the maximum scale (that of the whole patch) is the same for all permutations. However, the shape of the entropy function is quite different for each case.

The role of Step III, the inter-scale unpredictability measure W_D , is to weight the entropy value such that some permutations are preferred over others. It is

defined as the magnitude change of the PDF as a function of scale, therefore those orderings that are statistically self-dissimilar over scale are ranked higher than those that exhibit stationarity.

Figure 3(c) shows \mathcal{W}_D as a function of scale. It can be seen that the plot corresponding to the edge point has a much lower value than the one for the centre point at the selected scale value. In essence, it is a normalised measure of scale localisation. For example, in a noise image the pixel values are highly unpredictable at any one scale but over scale the statistics are stationary. However, a noise patch against a plain background would be salient due to the change in statistics.

In the continuous case, the saliency measure \mathcal{Y}_D , a function of scale s and position \mathbf{x} , is defined as:

$$\mathcal{Y}_D(\mathbf{s}_p, \mathbf{x}) \triangleq \mathcal{H}_D(\mathbf{s}_p, \mathbf{x}) \mathcal{W}_D(\mathbf{s}_p, \mathbf{x}) \quad (1)$$

i.e. for each point \mathbf{x} the set of scales \mathbf{s}_p , at which entropy peaks, is obtained, then the saliency is determined by weighting the entropy at these scales by \mathcal{W}_D . Entropy, \mathcal{H}_D , is given by:

$$\text{I} \quad \mathcal{H}_D(s, \mathbf{x}) \triangleq - \int p(I, s, \mathbf{x}) \log_2 p(I, s, \mathbf{x}) dI \quad (2)$$

where $p(I, s, \mathbf{x})$ is the probability density of the intensity I as a function of scale s and position \mathbf{x} . The set of scales \mathbf{s}_p is defined by:

$$\text{II} \quad \mathbf{s}_p \triangleq \left\{ s : \frac{\partial \mathcal{H}_D(s, \mathbf{x})}{\partial s} = 0, \frac{\partial^2 \mathcal{H}_D(s, \mathbf{x})}{\partial s^2} < 0 \right\} \quad (3)$$

The inter-scale saliency measure, $\mathcal{W}_D(s, \mathbf{x})$, is defined by:

$$\text{III} \quad \mathcal{W}_D(s, \mathbf{x}) \triangleq s \int \left| \frac{\partial}{\partial s} p(I, s, \mathbf{x}) \right| dI \quad (4)$$

In this paper, entropy is measured for the grey level image intensity but other attributes, e.g. colour or orientation, may be used instead; see [8] for examples.

This approach has a number of attractive properties. It offers a more general model of feature saliency and scale compared to conventional feature detection techniques. Saliency is defined in terms of spatial unpredictability; scale by the sampling window and its parameterisation. For example, a blob detector implemented using a convolution of multiple scale Laplacian-of-Gaussian (LoG) functions [14], whilst responding to a number of different feature shapes, maximally responds only to LoG function itself (or its inverse); in other words, it acts as a matched filter¹. Many convolution based approaches to feature detection exhibit the same bias, i.e. a preference towards certain features. This specificity has a detrimental effect on the quality of the features and scales selected. In

¹ This property is somewhat alleviated by the tendency of blurring to smooth image structures into LoG like functions.

contrast, the saliency approach responds equally to the LoG and all other permutations of its pixels provided that the constraint on $\mathcal{W}_D(s)$ is satisfied. This property enables the method to perform well over intra-class variations as is demonstrated in Section 4.

2.2 Affine Invariant Saliency

In the original formulation of [10], the method was invariant to the similarity group of geometric transformations and to photometric shifts. In this section, we develop the method to be fully affine invariant to geometric transformations. In principle, the modification is quite straightforward and may be achieved by replacing the circular sampling window by an ellipse: under an affine transformation, circles map onto ellipses. The scale parameter s is replaced by a vector $\mathbf{s} = (s, \rho, \theta)$, where ρ is the axis ratio and θ the orientation of the ellipse. Under such a scheme, the major and minor axes of the ellipse are given by $s/\sqrt{\rho}$ and $s\sqrt{\rho}$ respectively.

Increasing the dimensionality of the sampling window creates the possibility of degenerate cases. For example, in the case of a dark circle against a white background (see Figure 3(a)) any elliptical sampling window that contains an equal number of black and white pixels (\mathcal{H}_D constraint) but does not exclude any black pixels at the previous scale (\mathcal{W}_D constraint) will be considered equally salient. Such cases are avoided by requiring that the inter-scale saliency, \mathcal{W}_D , is smooth across a number of scales. A simple way to achieve this is to apply a 3-tap averaging filter to \mathcal{W}_D over scale.

2.3 Local Search

The complexity of a full search can be significantly reduced by adopting a local strategy in the spirit of [2,19,20]. Our approach is to start the search only at seeds points (positions and scales) found by applying the original similarity invariant search. Each seed circle is then locally adapted in order to maximise two criteria, \mathcal{H}_D (entropy) and \mathcal{W}_D (inter-scale saliency). \mathcal{W}_D is maximised when the ratio and orientation match that of the local image patch [9] at the correct scale, defined by a peak in \mathcal{H}_D . Therefore, we adopt an iterative refinement approach. The ratio and orientation are adjusted in order to maximise \mathcal{W}_D , then the scale is adjusted such that \mathcal{H}_D is peaked. The search is stopped when neither the scale nor shape change (or a maximum iteration count is exceeded).

The final set of regions are chosen using a greedy clustering algorithm which operates from the most salient feature down (highest value of \mathcal{Y}_D) and clusters together all features within the support region of the current feature. A global threshold on value or number is used.

The performance of this local method is compared to exhaustive search in Section 3.

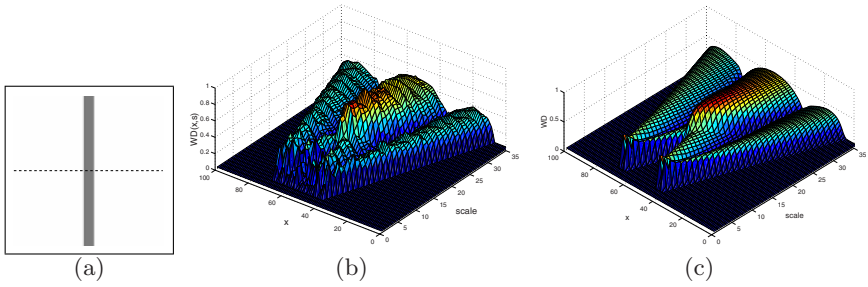


Fig. 4. \mathcal{W}_D as a function of x and scale for image shown in (a) at the y -position indicated by the dashed line using standard sampling (b), and anti-aliased sampling (c).

2.4 Anti-aliased Sampling

The simplest method for estimating local PDFs from images is to use histogramming over a local neighbourhood, for example a circular region; pixels inside the region are counted whilst those outside are not. However, this binary approach gives rise to step changes in the histogram as the scale is increased. \mathcal{W}_D is especially sensitive to this since it measures the difference between two concentric sampling windows. For example, Figure 4(b) shows the variation of \mathcal{W}_D as a function of x and scale for the image shown in 4(a). The surface is taken at a point indicated by the dashed line. Somewhat surprisingly the surface is highly irregular and noisy even for this ideal noise-free image, consequently, so is the saliency space. Intuitively, the solution to this problem lies with a smoother transition between the pixels that are included in the histogram and the ones that are not.

The underlying problem is, in fact, an instance of aliasing. Restated from a sampling perspective, the binary representation of the window is sampling without pre-filtering. Evidently, this results in severe aliasing. This problem has long been recognised in the Computer Graphics community and numerous methods have been devised to better represent primitives on a discrete display [5].

To overcome this problem we use a smooth sampling window (i.e. a filtered version of the ideal sampling window). However, in contrast to the CG application, here, the window weights the contributions of the pixels to the histogram not the pixel values themselves; pixels near the edge contribute less to the count than ones near the centre. It does not blur the image.

Griffin [6] and Koenderink and van Doorn [12] have suggested weighting histogram counts using a Gaussian window, but not in relation to anti-aliasing. However, for our purposes, the Gaussian poorly represents the statistics of the underlying pixels towards the edges due to the slow drop-off. Its long tails cause a slow computation since more pixels will have to be considered and also results in poor localisation. The traditional ‘pro-Gaussian’ arguments do not seem appropriate here.

Analytic solutions for the optimal sampling window are, in theory at least, possible to obtain. However, empirically we have found the following function works well:

$$SW(z) = \frac{1}{1 + \left(\frac{z}{s}\right)^n} \quad z = \sqrt{\left(\frac{x'}{\sqrt{\rho}}\right)^2 + (y'\sqrt{\rho})^2}. \quad (5)$$

with $n = 42$ where $x' = x \cos \theta + y \sin \theta$ and $y' = y \cos \theta - x \sin \theta$ achieves the desired rotation. We truncate for small values of $SW(z)$. This sampling window gives scalar values as a function of distance, z , from the window centre, which are used to build the histogram. Figure 4(c) shows the same slice through \mathcal{W}_D space but generated using Equation 5 for the sampling weights. Further implementation details and analysis may be found in [9,10].

3 Performance under Viewpoint Variations

The objective here is to determine the extent to which detected regions commute with viewpoint. This is an example of the global transformation requirement discussed in the introduction.

For these experiments, we follow the testing methodology proposed in [18, 19]. The method is applied to an image set² comprising different viewpoints of the same (largely planar) scene for which the inter-image homography is known. Repeatability is determined by measuring the area of overlap of corresponding features. Two features are deemed to correspond if their projected positions differ by less than 1.5 pixels. Results are presented in terms of error in overlapping area between two ellipses μ_a, μ_b :

$$\epsilon_S = 1 - \frac{\mu_a \cap (A^T \mu_b A)}{\mu_a \cup (A^T \mu_b A)} \quad (6)$$

where A defines a locally linearized affine transformation of the homography between the two images and $\mu_a \cap (A^T \mu_b A)$ and $\mu_a \cup (A^T \mu_b A)$ represent the area of intersection and union of the ellipses respectively.

Figure 5(a) shows the repeatability performance as a function of viewpoint of three variants of the affine invariant salient region detector: exhaustive search without anti-aliasing (FS Affine ScaleSal), exhaustive search with anti-aliasing (AA FS Affine ScaleSal), and local search with anti-aliasing (AA LS Affine ScaleSal). The performance is compared to the detector of Mikolajczyk and Schmid [19], denoted Affine MSHar. Results are shown for $\epsilon_S < 0.4$.

It can be seen that the full search Affine Saliency and Affine MSHar features have a similar performance over the range of viewpoints. However, from 40° the anti-aliased sampling provides some gains, though curiously diminishes performance at 20°. The local search anti-aliased Affine Saliency performs reasonably well compared to the full search methods but of course takes a fraction of the time to compute.

² Graffiti6 from <http://www.inrialpes.fr/lear/people/Mikolajczyk/>

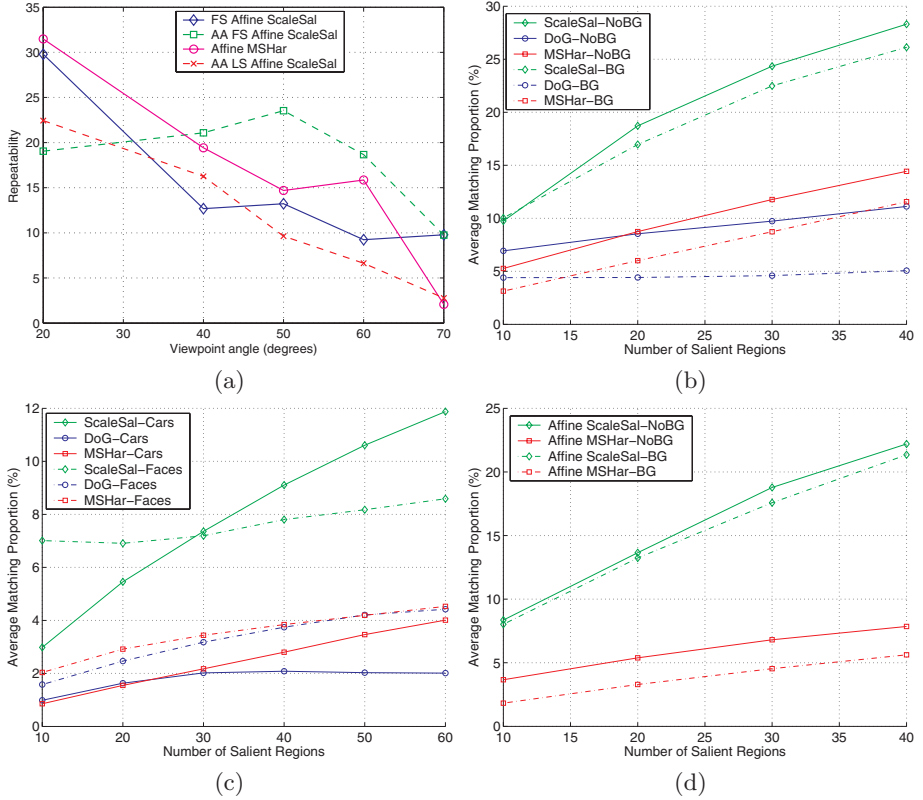


Fig. 5. Repeatability results under (a) viewpoint changes, (b,d) background perturbations and intra-class variations for Bike images, (c) intra-class variations for car and face images. Plots (b,c) are for similarity invariant and (d) for affine invariant detectors.

4 Performance under Intra-class Variation and Image Perturbations

The aim here is to measure the performance of a region detector under intra-class variations and image perturbations – the other two requirements specified in the introduction. In the following subsections we develop this measure and then compare performance of the salient region detector to other region operators. In these experiments we used similarity invariant versions of three detectors: similarity Saliency (ScaleSal), Difference-Of-Gaussian (DoG) blob detector [16] and the multi-scale Harris (MSHar) with Laplacian scale selection — this is Affine MSHar without the affine adaptation [19]. We also used affine invariant detectors Affine ScaleSal and Affine MSHar. An affine invariant version of the DoG detector was not available.

4.1 The Performance Measure

We will discuss first measuring repeatability over intra-class variation. Suppose we have a set of images of the same object class, e.g. motorbikes. A region detection operator which is unaffected by intra-class variation will reliably select regions on *corresponding parts* of all the objects, say the wheels, engine or seat for motorbikes. Thus, we assess performance by measuring the (average) number of correct correspondences over the set of images.

The question is: what constitutes a correct corresponding region? To determine this, we use a proxy to the true intra-class transformation by assuming that an affinity approximately maps one imaged object instance to another. The affinities are estimated here by manually clicking on corresponding points in each image, e.g. for motorbikes the wheels and seat/petrol tank join. We consider a region to match if it fulfils three requirements: its position matches within 10 pixels; its scale is within 20% and normalised mutual information³ between the appearances is > 0.2 . For the affine invariant detectors, the scale test is replaced with the overlap error, $\epsilon_s < 0.4$ (Eq. 6), and the mutual information is applied to elliptical patches transformed to circles. These are quite generous thresholds since the objects *are* different and the geometric mapping approximate.

In detail we measure the average correspondence score S as follows. N regions are detected on each image of the M images in the dataset. Then for a particular reference image i the correspondence score S_i is given by the proportion of corresponding to detected regions for all the other images in the dataset, i.e.:

$$S_i = \frac{\text{Total number of matches}}{\text{Total number of detected regions}} = \frac{N_M^i}{N(M-1)} \quad (7)$$

The score S_i is computed for $M/2$ different selections of the reference image, and averaged to give S . The score is evaluated as a function of the number of detected regions N . For the DoG and MSHar detectors the features are ordered on Laplacian (or DoG) magnitude strength, and the top N regions selected.

In order to test insensitivity to image perturbation the data set is split into two parts: the first contains images with a uniform background and the second, images with varying degrees of background clutter. If the detector is robust to background clutter then the average correspondence score S should be similar for both subsets of images.

4.2 Intra-class Variation Results

The experiments are performed on three separate data-sets, each containing different instances from an object class: 200 images from Caltech Motorbikes (Side), 200 images from Caltech Human face (Front), and all 126 Caltech Cars (Rear) images. Figure 6 shows examples from each data set⁴.

³ $MI(A, B) = 2(H(A) + H(B) - H(A, B))/(H(A) + H(B))$

⁴ Available from <http://www.robots.ox.ac.uk/~vgg/data/>.



Fig. 6. Example images from (a) the two parts of the Caltech motorbike data set without background clutter (top) and with background clutter (bottom), and (b) Caltech cars (top) and Caltech faces (bottom).

The average correspondence score S results for the similarity invariant detectors are shown in Figure 5(b) and (c). Figure 5(d) shows the results for the affine detectors on the motorbikes. For all three data sets and at all thresholds the best results are consistently obtained using the saliency detector. However, the repeatability for all the detectors is lower for the face and cars compared to the motorbike case. This could be due to the appearances of the different object classes; motorbikes tend to appear more complex than cars and faces.

Figure 7 shows smoothed maps of the locations at which features were *detected* in all 200 images in the motorbike image set. All locations have been back projected onto a reference image. Bright regions are those at which detections are more frequent. The map for the saliency detector indicates that most detections are near the object with a few high detection points near the engine, seats wheel centres, headlamp. In contrast, the DoG and MSHar maps show a much more diffuse pattern over the entire area caused by poor localisation and false responses to background clutter.

4.3 Image Perturbation Results

The motorbike data set is used to assess insensitivity to background clutter. There are 84 images with a uniform background, and 116 images with varying degrees of background clutter; see Figure 6(a).

Figure 5(b) shows separate plots for motorbike images with and without background clutter at $N=10$ to 40. The saliency detector finds, on average, approximately 25% of 30 features within the matching constraints; this corresponds to about 7 features per image on average. In contrast, the MSHar and DoG detectors select 2-3 object features per image at this threshold. Typical examples of the matched regions selected by the saliency detector on this data set are shown in Figure 4.3.

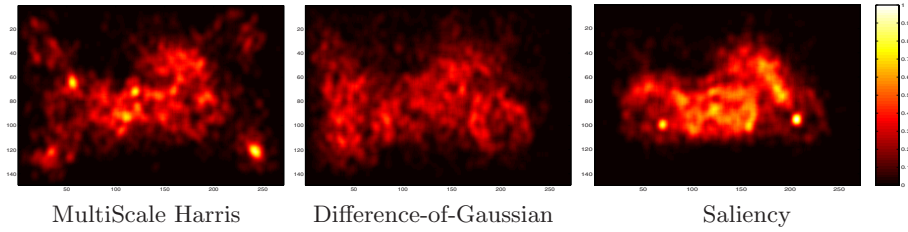


Fig. 7. Smoothed map of the detected features over all 200 images in the motorbike set back projected onto one image. The colour indicates the normalised number of detections in a given area (white is highest). Note the relative ‘tightness’ of the bright areas of the saliency detector compared to the DoG and MSHarr.

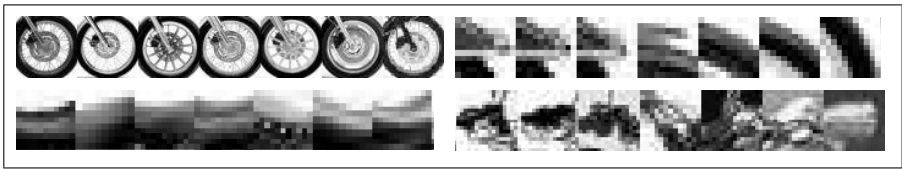


Fig. 8. Examples of the matched regions selected by the similarity saliency detector from the motorbike images: whole front wheels; front mud-guard/wheel corner; seat; headlamp.

There is also a marked difference in the way the various detectors are affected by clutter. It has little effect on the ScaleSal detector whereas it significantly reduces the DoG performance and similarly that of MSHar. Similar trends are obtained for the affine invariant detectors applied to the motorbikes images, shown in Figure 5(d).

Local perturbations due to changes in the scene configuration, background clutter or changes within in the object itself can be mitigated by ensuring compact support of any probing elements. Both the DoG and MSHar methods rely on relatively large support windows which cause them to be affected by non-local changes in the object and background; compare the two cluttered and uncluttered background results for the motorbike experiments.

There may be several other relevant factors. First, both the DoG and MSHar methods blur the image, hence causing a greater degree of similarity between objects and background. Second, in most images the objects of interest tend to be in focus while backgrounds are out of focus and hence blurred. Blurred regions tend to exhibit slowly varying statistics which result in a relatively low entropy and inter-scale saliency in the saliency detector. Third, the DoG and MSHar methods define saliency with respect to specific properties of the local surface geometry. In contrast, the saliency detector uses a much broader definition.

5 Discussion and Future Work

In this paper we have presented a new region detector which is comparable to the state of the art [19,20] in terms of co-variance with viewpoint. We have also demonstrated that it has superior performance on two further criteria: robustness to image perturbations, and repeatability under intra-class variability. The new detector extends the original method of Kadir and Brady to affine invariance; we have developed a properly anti-aliased implementation and a fast optimisation based on a local search.

We have also proposed a new methodology to test detectors under intra-class variations and background perturbations. Performance under this extended criterion is important for many applications, for example part detectors for object recognition.

The intra-class experiments demonstrate that defining saliency in the manner of the saliency detector is, on average, a better search heuristic than the other region detectors tested on at least the three data sets used here.

It is interesting to consider how the design of feature detectors affects performance. Many global effects, such as viewpoint, scale or illumination variations can be modelled mathematically and as such can be tackled directly provided the detector also lends itself to such analysis. Compared to the diffusion-based scale-spaces, relatively little is currently known about the properties of spaces generated by statistical methods such as that described here. Further investigation of its properties seems an appealing line of future work.

We plan to compare the saliency detector to other region detection approaches which are not based on filter response extrema such as [17,22]

Acknowledgements. We would like to acknowledge Krystian Mikolajczyk for supplying the MSHar and embedded (David Lowe) DoG feature detector code. Thanks to Mark Everingham and Josef Sivic for many useful discussions and suggestions. This work was funded by EC project CogViSys.

References

1. S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proc. European Conf. Computer Vision*, pages 113–130, 2002.
2. A. Baumberg. Reliable feature matching across widely separated views. In *Proc. Computer Vision Pattern Recognition*, pages 774–781, 2000.
3. E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *Proc. European Conf. Computer Vision*, pages 109–124, 2002.
4. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. Computer Vision Pattern Recognition*, pages II: 264–271, 2003.
5. J.A. Foley and A. Van Dam. *Fundamentals of Interactive Computer Graphics*. Addison-Wesley, 1982.
6. L.D. Griffin. Scale-imprecision space. *Image and Vision Computing*, 15:369–398, 1997.

7. C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. Alvey Vision Conf.*, pages 189–192, 1988. Manchester.
8. T. Kadir. *Scale, Saliency and Scene Description*. PhD thesis, University of Oxford, 2002.
9. T. Kadir, D. Boukerroui, and J.M. Brady. An analysis of the scale saliency algorithm. Technical Report OUEL No: 2264/03, University of Oxford, 2003.
10. T. Kadir and J.M. Brady. Scale, saliency and image description. *Intl. J. of Computer Vision*, 45(2):83–105, 2001.
11. J.J. Koenderink and A.J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 63:291–297, 1987.
12. J.J. Koenderink and A.J. van Doorn. The structure of locally orderless images. *Intl. J. of Computer Vision*, 31(2/3):159–168, 1999.
13. S. Kok-Wiles, M. Brady, and R. Highnam. Comparing mammogram pairs for the detection of lesions. In *Proc. Intl. Workshop on Digital Mammography*, pages 103–110, 1998.
14. T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *Intl. J. of Computer Vision*, 11(3):283–318, 1993.
15. T. Lindeberg and B.M. ter Haar Romeny. Linear scale-space: I. basic theory, II. early visual operations. In B.M. ter Haar Romeny, editor, *Geometry-Driven Diffusion*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.
16. D.G. Lowe. Object recognition from local scale-invariant features. In *Proc. Intl. Conf. on Computer Vision*, pages 1150–1157, 1999.
17. J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. British Machine Vision Conf.*, pages 384–393, 2002.
18. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. Intl. Conf. on Computer Vision*, 2001.
19. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. European Conf. Computer Vision*, 2002.
20. F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or “How do I organize my holiday snaps?”. In *Proc. European Conf. Computer Vision*, pages 414–431, 2002.
21. C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.
22. T. Tuytelaars and L. Van Gool. Wide baseline stereo based on local, affinely invariant regions. In *Proc. British Machine Vision Conf.*, pages 412–422, 2000.
23. M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proc. European Conf. Computer Vision*, June 2000.

A Visual Category Filter for Google Images

R. Fergus¹, P. Perona², and A. Zisserman¹

¹ Dept. of Engineering Science,
University of Oxford, Parks Road,
Oxford, OX1 3PJ, UK.

{fergus,az}@robots.ox.ac.uk

² Dept. of Electrical Engineering,
California Institute of Technology,
MC 136-93, Pasadena, CA 91125, U.S.A.
perona@vision.caltech.edu

Abstract. We extend the *constellation model* to include heterogeneous parts which may represent either the appearance or the geometry of a region of the object. The parts and their spatial configuration are learnt simultaneously and automatically, without supervision, from cluttered images.

We describe how this model can be employed for ranking the output of an image search engine when searching for object categories. It is shown that visual consistencies in the output images can be identified, and then used to rank the images according to their closeness to the visual object category.

Although the proportion of good images may be small, the algorithm is designed to be robust and is capable of learning in either a totally unsupervised manner, or with a very limited amount of supervision.

We demonstrate the method on image sets returned by Google's image search for a number of object categories including bottles, camels, cars, horses, tigers and zebras.

1 Introduction

Just type a few keywords into the Google image search engine, and hundreds, sometimes thousands of pictures are suddenly available at your fingertips. As any Google user is aware, not all the images returned are related to the search. Rather, typically more than half look completely unrelated; moreover, the useful instances are not returned first – they are evenly mixed with unrelated images. This phenomenon is not difficult to explain: current Internet image search technology is based upon words, rather than image content – the filename of the image and text near the image on a web-page [4]. These criteria are effective at gathering quickly related images from the millions on the web, but the final outcome is far from perfect.

We conjecture that, even without improving the search engine per se, one might improve the situation by measuring ‘visual consistency’ amongst the images that are returned and re-ranking them on the basis of this consistency, so increasing the fraction of good images presented to the user within the first few web pages. This conjecture stems from the observation that the images that are related to the search typically are

visually similar, while images that are unrelated to the search will typically look different from each other as well.

How might one measure ‘visual consistency’? One approach is to regard this problem as one of probabilistic modeling and robust statistics. One might try and fit the data (the mix of images returned by Google) with a parametrized model which can accommodate the within-class variation in the requested category, for example the various shapes and labels of bottles, while rejecting the outliers (the irrelevant images). Learning a model of the category under these circumstances is an extremely challenging task. First of all: even objects within the same category do look quite different from each other. Moreover, there are the usual difficulties in learning from images such as lighting and viewpoint variations (scale, foreshortening) and partial occlusion. Thirdly, and most importantly, in the image search scenario the object is actually only present in a sub-set of the images, and this sub-set (and even its size) is unknown.

While methods exist to model object categories [9,13,15], it is essential that the approach can learn from a contaminated training set with a minimal amount of supervision. We therefore use the method of Fergus *et al.* [10], extending it to allow the parts to be heterogeneous, representing a region’s appearance or geometry as appropriate. The model and its extensions are described in section 2. The model was first introduced by Burl *et al.* [5]. Weber *et al.* [23] then developed an EM-based algorithm for training the model on cluttered datasets with minimal supervision. In [10] a probabilistic representation for part appearance was developed; the model made scale invariant; and both appearance and shape learnt simultaneously.

Other approaches to this problem [7,19] use properties of colour or texture histograms. While histogram approaches have been successful in Content Based Image Retrieval [2,12,21], they are unsuitable for our task since the within-class returns vary widely in colour and texture.

We explore two scenarios: in the first the user is willing to spend a limited amount of time (e.g. 20-30 seconds) picking a handful of images of which they want more examples (a simple form of relevance feedback [20]); in the second the user is impatient and there is no human intervention in the learning (i.e. it is completely unsupervised).

Since the model only uses visual information, a homonymous category (one that has multiple meanings, for example “chips” would return images of both “French fries” and “microchips”) pose problems due to multiple visual appearances. Consequently we will only consider categories with one dominant meaning in this paper. The algorithm only requires images as its input, so can be used in conjunction with any existing search engine. In this paper we have chosen to use Google’s image search.

2 The Model

In this section we give an overview of our previously developed method [10], together with the extension to heterogeneous parts.

An object model consists of a number of parts which are spatially arranged over the object. A part here may be a patch of pixels or a curve segment. In either case, a part is represented by its intrinsic description (appearance or geometry), its scale relative to the model, and its occlusion probability. The overall model shape is represented by the

mutual position of the parts. The entire model is generative and probabilistic, so part description, scale, model shape and occlusion are all modeled by probability density functions, which are Gaussians.

The process of learning an object category is one of first detecting features with characteristic scales, and then estimating the parameters of the above densities from these features, such that the model gives a maximum-likelihood description of the training data. Recognition is performed on a query image by again first detecting features (and their scales), and then evaluating the features in a Bayesian manner, using the model parameters estimated in the learning.

2.1 Model Structure Overview

A model consists of P parts and is specified by parameters θ . Given N detected features with locations \mathbf{X} , scales \mathbf{S} , and descriptions \mathbf{D} , the likelihood that an image contains an object is assumed to have the following form:

$$p(\mathbf{X}, \mathbf{S}, \mathbf{D} | \theta) = \sum_{\mathbf{h} \in H} \underbrace{p(\mathbf{D} | \mathbf{X}, \mathbf{S}, \mathbf{h}, \theta)}_{\text{Part Description}} \underbrace{p(\mathbf{X} | \mathbf{S}, \mathbf{h}, \theta)}_{\text{Shape}} \underbrace{p(\mathbf{S} | \mathbf{h}, \theta)}_{\text{Rel. Scale}} \underbrace{p(\mathbf{h} | \theta)}_{\text{Other}}$$

where the summation is over allocations, \mathbf{h} , of parts to features. Typically a model has 5-7 parts and there will be around thirty features of each type in an image.

Similarly it is assumed that non-object background images can be modeled by a likelihood of the same form with parameters θ_{bg} . The decision as to whether a particular image contains an object or not is determined by the likelihood ratio:

$$R = \frac{p(\mathbf{X}, \mathbf{S}, \mathbf{D} | \theta)}{p(\mathbf{X}, \mathbf{S}, \mathbf{D} | \theta_{bg})} \quad (1)$$

The model, at both the fitting and recognition stages, is scale invariant. Full details of the model and its fitting to training data using the EM algorithm are given in [10], and essentially the same representations and estimation methods are used.

2.2 Heterogeneous Parts

Existing approaches to recognition learn a model based on a single type of feature (e.g. image patches [3, 16], texture regions [18] or Haar wavelets [22]). However, the different visual nature of objects means that this is limiting. For some objects, like wine bottles, the essence of the object is captured far better with geometric information (the outline) rather than by patches of pixels. Of course, the reverse is true for many objects, like humans faces. Consequently, a flexible visual recognition system must have multiple feature types. The flexible nature of the constellation model makes this possible. As the description densities of each part are independent, each can use a different type of feature.

In this paper, only two types of features are included, although more can easily be added. The first consists of regions of pixels, this being the feature type used previously; the second consists of curve segments. Figure 1 illustrates these features on two typical images. These feature are complementary: one represents the *appearance* of object patches, the other represents the object *geometry*.

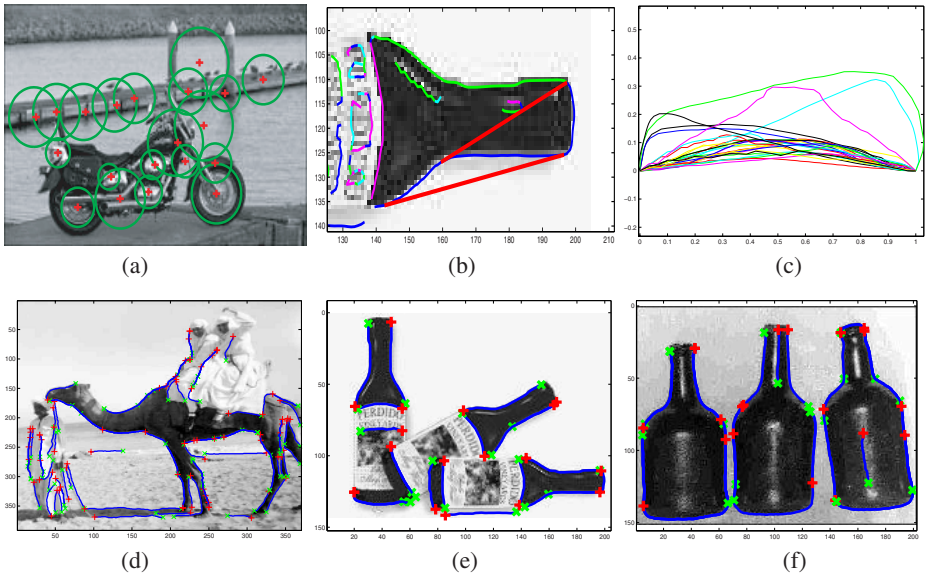


Fig. 1. (a) Sample output from the region detector. The circles indicate the scale of the region. (b) A long curve segment being decomposed at its bitangent points. (c) Curves within the similarity-invariant space - note the clustering. (d), (e) & (f) show the curve segments identified in three images. The green and red markers indicate the start and end of the curve respectively

2.3 Feature Detection

Pixel patches. Kadir and Brady's interest operator [14] finds regions that are salient over both location and scale. It is based on measurements of the grey level histogram and entropy over the region. The operator detects a set of circular regions so that both position (the circle centre) and scale (the circle radius) are determined, along with a saliency score. The operator is largely invariant to scale changes and rotation of the image. For example, if the image is doubled in size then a corresponding set of regions will be detected (at twice the scale). Figure 1(a) shows the output of the operator on a sample image.

Curve segments. Rather than only consider very local spatial arrangements of edge points (as in [1]), extended edge chains are used, detected by the Canny edge operator [6]. The chains are then segmented into segments between bitangent points, i.e. points at which a line has two points of tangency with the curve. Figure 1(b) shows an example.

This decomposition is used for two reasons: first, bitangency is covariant with projective transformations. This means that for near planar curves the segmentation is invariant to viewpoint, an important requirement if the same, or similar, objects are imaged at different scales and orientations. Second, by segmenting curves using a bi-local property interesting segments can be found consistently despite imperfect edgel data.

Bitangent points are found on each chain using the method described in [17]. Since each pair of bitangent points defines a curve which is a sub-section of the chain, there

may be multiple decompositions of the chain into curved sections as shown in figure 1(b). In practice, many curve segments are straight lines (within a threshold for noise) and these are discarded as they are intrinsically less informative than curves. In addition, the entire chain is also used, so retaining convex curve portions.

2.4 Feature Representation

The feature detectors gives patches and curves of interest within each image. In order to use them in our model their properties are parametrized to form $\mathbf{D} = [\mathbf{A}, \mathbf{G}]$ where \mathbf{A} is the appearance of the regions within the image, and \mathbf{G} is the shape of the curves within each image.

Region representation. As in [10], once the regions are identified, they are cropped from the image and rescaled to a smaller, 11×11 pixel patch. The dimensionality is then reduced using principal component analysis (PCA). In the learning stage, patches from all images are collected and PCA performed on them. Each patch's appearance is then a vector of the coordinates within the first 15 principal components, so giving \mathbf{A} .

Curve representation. Each curve is transformed to a canonical position using a similarity transformation such that it starts at the origin and ends at the point $(1, 0)$. If the curve's centroid is below the x -axis then it is flipped both in the x -axis and the line $y = 0.5$, so that the same curve is obtained independent of the edgel ordering. The y value of the curve in this canonical position is sampled at 13 equally spaced x intervals between $(0, 0)$ and $(1, 0)$. Figure 1(c) shows curve segments within this canonical space. Since the model is not orientation-invariant, the original orientation of the curve is concatenated to the 13-vector for each curve, giving a 15-vector (for robustness, orientation is represented as a normalized 2-vector). Combining the 15-vectors from all curves within the image gives \mathbf{G} .

2.5 Model Structure and Representation

The descriptors are modelled by the $p(\mathbf{D}|\mathbf{X}, \mathbf{S}, \mathbf{h}, \theta)$ likelihood term. Each part models either curves or patches and this allocation is made beforehand. \mathbf{h} picks a feature for each part from \mathbf{A} or \mathbf{G} (as appropriate) and is then modelled by a 15 dimensional Gaussian (note that both curves and patches are represented by a 15-vector). This Gaussian will hopefully find a cluster of curves/patches close together in the space, corresponding to similar looking curves or patches across images. The relative locations of the model parts are modelled by $p(\mathbf{X}|\mathbf{S}, \mathbf{h}, \theta)$ – which is a joint Gaussian density over all parts. Again, \mathbf{h} allocates a feature to each part. The location of curve is taken as its centroid. The location of a patch is its region centre. For the relative scale term, $p(\mathbf{S}|\mathbf{h}, \theta)$ – again a Gaussian, the length of the curve and the radius of a patch region is taken as being the scale for a curve/patch.

3 Method

In this section the experimental implementation is described: the gathering of images, feature detection, model learning and ranking. The process will be demonstrated on the “bottles” category .

3.1 Image Collection

For a given keyword, Google’s image search¹ was used to download a set of images. Images outside a reasonable size range (between 100 and 600 pixels on the major axis) were discarded. A typical image search returned in the region of 450-700 usable images. A script was used to automate the procedure. For assessment purposes, the images returned were divided into 3 distinct groups (see fig. 2):

1. **Good images:** these are good examples of the keyword category, lacking major occlusion, although there may be a variety of viewpoints, scalings and orientations.
2. **Intermediate images:** these are in some way related to the keyword category, but are of lower quality than the good images. They may have extensive occlusion; substantial image noise; be a caricature or cartoon of the category; or the category is rather insignificant in the image, or some other fault.
3. **Junk images:** these are totally unrelated to the keyword category.

Additionally, a dataset consisting entirely of junk images was collected, by using the keyword “things”. This background dataset is used in the unsupervised learning procedure.

The algorithm was evaluated on ten datasets gathered from Google: bottles, camel, cars, coca cola, horses, leopards, motorbike, mugs, tiger and zebra. It is worth noting that the inclusion or exclusion of an “s” to the keyword can make a big difference to the images returned. The datasets are detailed in Table 1.

Table 1. Statistics of the datasets as returned by Google.

Dataset	Bottles	Camel	Cars	Coca-cola	Horses	Leopards	Motorbike	Mugs	Tiger	Zebra	Things
Total size of dataset	700	700	448	500	600	700	500	600	642	640	724
% Good images	41	24	30	17	21	49	25	50	35	44	n.a.
% Intermediate images	26	27	18	12	25	33	16	9	24	33	n.a.
% Junk images	33	49	52	71	54	18	59	41	41	24	n.a.

3.2 Image Re-ranking

Feature detection. Each image is converted to greyscale, since colour information is not used in the model. Curves and regions of interest are then found within the image, using exactly the same settings for all datasets. This produces **X**, **D** and **S** for use in learning or recognition. The 25 regions with the highest saliency, and 30 curves with the longest length are used from each image.

¹ <http://www.google.com/imghp>. Date of collection: Jan. 2003. As we write (Feb. 2004) we notice that Google’s precision-recall curves have improved during the last 12 months.



Fig. 2. Images of bottles. (a) the first 25 images returned by Google. The coloured dot in the bottom right hand corner indicates the ground truth category of the image: good (green); intermediate (yellow) or junk (red). (b) the 10 hand selected images used in the supervised experiments.

Model Learning. The learning process takes one of two distinct forms: unsupervised learning and limited supervision:

- **Unsupervised learning:** In this scenario, a model is learnt using all images in the dataset. No human intervention is required in the process.
- **Learning with limited supervision:** An alternative approach is to use relevance-feedback. The user picks 10 or so images that are close to the image he/she wants, see figure 2(b) for examples for the bottles category. A model is learnt using these images.

In both approaches, the learning task takes the form of estimating the parameters θ of the model discussed above. The goal is to find the parameters $\hat{\theta}_{ML}$ which best explain the data $\mathbf{X}, \mathbf{D}, \mathbf{S}$ from the chosen training images (be it 10 or the whole dataset), i.e. maximise the likelihood: $\hat{\theta}_{ML} = \arg \max_{\theta} p(\mathbf{X}, \mathbf{D}, \mathbf{S} | \theta)$. For the 5 part model used in the experiments, there are 243 parameters. In the supervised learning case, the use of only 10 training images is a compromise between the number the user can be expected to pick and the generalisation ability of the model. The model is learnt using the EM algorithm as described in [10]. Figure 3 shows a curve model and a patch model trained from the 10 manually selected images of bottles.

Re-ranking. Given the learnt model, the likelihood ratio (eqn. 1) for each image is computed. This likelihood ratio is then used to rank all the images in the dataset. Note that in the supervised case, the 10 images manually selected are excluded from the ranking.

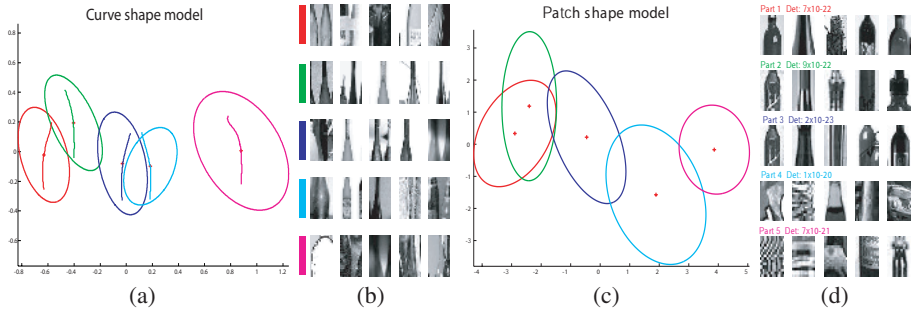


Fig. 3. Models of bottles. (a) & (b): Curve model. (c) & (d): Patch model. (a) The spatial layout of the curve model with mean curves overlaid. The X and Y axes are in arbitrary units since the model is scale-invariant. The ellipses indicate variance in relative location. (b) Patch of images selected by curve features from high scoring hypotheses. (c) Spatial layout for patch model. (d) Sample patches closest to mean of appearance density. Both models pick out bottle necks and bodies with the shape model capturing the side-by-side arrangement of the bottles.

Speed considerations. If this algorithm is to be of practical value, it must be fast. Once images have been preprocessed, which can be done off-line, a model can be learnt from 10 images in around 45 seconds and the images in the dataset re-ranked in 4 – 5 seconds on a 2 Ghz processor.

3.3 Robust Learning in the Unsupervised Case

We are attempting to learn a model from a dataset which contains valid data (the good images) but also *outliers* (the intermediate and junk images), a situation faced in the area of robust statistics. One approach would be to use all images for training and rely on the models' occlusion term to account for the small portion of valid data. However, this requires an accurate modelling of image clutter properties and reliable convergence during learning. An alternative approach, we adapt a robust fitting algorithm, RANSAC [11], to our needs. A large number of models are trained (~ 100), each one using a set of randomly drawn images sufficient to train a model (10 in this case). The intuition is that at least one of these will be trained on a higher than average proportion of good images, so will be a good classifier. The challenge is to find a robust unsupervised scoring function that is highly correlated to the underlying classification performance. The model with the highest score is then picked as model to perform the re-ranking of the dataset.

Our novel scoring approach uses a second set of images, consisting entirely of irrelevant images, the aforementioned background dataset. Thus there are now two datasets: (a) the one to be ranked (consisting of a mixture of junk and good images) and (b) the background dataset. Each model evaluates the likelihood of images from both datasets and a differential ranking measure is computed between them. In this instance, we compute the area under a recall-precision curve (RPC) between the two datasets. In our experiments we found a good correlation between this measure and the ground truth RPC precision: the final model picked was consistently in the top 15% of models, as demonstrated in figs. 4(c) & (d).

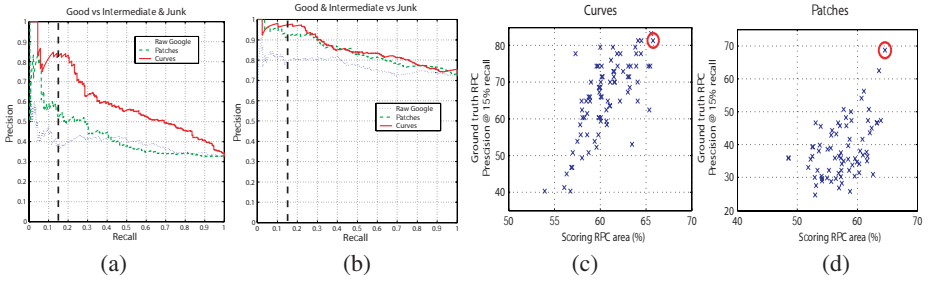


Fig. 4. (a) & (b) Recall-Precision curves computed using ground truth for the supervised models in figure 3. In (a), the good images form the positive set and the intermediate and junk images form the negative one. In (b), good and intermediate images form the positive set and junk images, the negative one. The dotted blue line is the curve of the raw Google images (i.e. taken in the order they are presented to the user). The solid red line shows the performance of the curve model and the dashed green line shows the performance of the patch model. As most users will only look at the first few pages of returned results, the interesting area of the plots is the left-hand side of the graph, particularly around a recall of 0.15 (as indicated by the vertical line). In this region, the curve model clearly gives an improvement over both the raw images and the patch model (as predicted by the variance measure). (c) & (d): Scatter plots showing the scoring RPC area versus ground truth RPC area for curve and patch models respectively in the unsupervised learning procedure. Each point is a model learnt using the RANSAC-style unsupervised learning algorithm. The model selected for each feature type is indicated by the red circle. Note that in both plots it is amongst the best few models.

3.4 Selection of Feature Type

For each dataset in both the supervised and unsupervised case, two different models are learnt: one using only patches and another using only curves. A decision must be made as to which model should give the final ranking that will be presented to the user. This is a challenging problem since the models exist in different spaces, so their likelihoods cannot be directly compared. Our solution is to compare the variance of the unsupervised models' scoring function. If a feature type is effective then a large variance is expected since a good model will score much better than a mediocre one. However, an inappropriate feature type will be unable to separate the data effectively, no matter which training images were used, meaning all scores will be similar.

Using this approach, the ratio of the variance of the RANSAC curve and patch models is compared to a threshold (fixed for all datasets) and a selection of feature type is made. This selection is then used for both the unsupervised and supervised learning cases. Figure 5 shows the first few re-ranked images of the bottles dataset, using the model chosen - in this case, curves.

4 Results

Two series of experiments were performed: the first used the supervised learning method while the second was completely unsupervised. In both sets, the choice between curves and patches was made automatically. The results of the experiments are summarised in table 2.



Fig. 5. Re-ranked bottle images. The dot in the bottom right corner shows the label of the image. The thin magenta curves on each image show the curve segments detected. The best hypothesis is also highlighted with thick coloured lines. The duplicate images present in the dataset are the reason that some of the 10 training images appear in the figure. Notice that the model seems to pick up the neck of the bottles, with its distinctive curvature. These images clearly contain more bottles than those of figure 2.

Table 2. Summary of results: Precision at 15% recall - equivalent to around two web-pages worth of images. Good images vs. intermediate & junk. The second row gives raw Google output precision. Rows 3 & 4 give results of supervised learning, using 10 handpicked images. Rows 5 & 6 give results of unsupervised RANSAC-style learning. Rows 7 & 8 are included to show the comparison of the RANSAC approach to unsupervised learning using all images in the dataset. Bold indicates the automatically selected model. For the forms of learning used (supervised and RANSAC-style unsupervised), this model selection is correct 90% of the time. The final column gives the average precision across all datasets, for the automatically chosen feature type.

Dataset	Bottles	Camel	Cars	Coca-cola	Horses	Leopards	Motorbike	Mugs	Tiger	Zebra	Average
Raw Google	39.3	36.1	31.7	41.9	31.1	46.8	48.7	84.9	30.5	51.9	44.3
10 images (Curves)	82.9	80.0	78.3	35.3	28.3	39.5	48.6	75.0	43.8	74.1	65.9
10 images (Patches)	52.3	68.6	47.4	54.5	23.6	69.0	42.5	55.7	72.7	74.1	
RANSAC unsupervised-Curves	81.4	78.8	69.0	29.5	25.0	41.5	61.3	68.2	43.4	71.2	58.9
RANSAC unsupervised-Patches	68.6	48.7	42.6	26.0	25.0	50.0	20.4	66.7	58.9	54.5	
All images unsupervised-Curves	76.1	81.2	41.7	43.3	23.2	51.0	34.5	76.3	44.0	64.6	52.9
All images unsupervised-Patches	35.0	27.4	44.4	23.6	22.4	55.4	17.9	62.5	53.2	50.0	

4.1 Supervised Learning

The results in table 2 show that the algorithm gives a marked improvement over the raw Google output in 7 of the 10 datasets. The evaluation is a stringent one, since the model must separate the good images from the intermediate and junk, rather than just separating the good from the junk. The curve features were used in 6 instances, as compared to 4 for patches. While curves would be expected to be preferable for categories such as bottles, their marked superiority on the cars category, for example, is surprising. It can be explained by the large variation in viewpoint present in the images. No patch features could be found that were stable across all views, whereas long horizontal curves in close proximity were present, regardless of the viewpoint and these were used by the model, giving a good performance. Another example of curves being unexpectedly effective, is on the camel dataset, as shown in figure 6. Here, the knobbly knees and legs of the camel

are found consistently, regardless of viewpoint and clutter, so are used by the model to give a precision (at 15% recall) over twice that of the raw Google images. The failure to improve Google's output on 3 of the categories (horses, motorbikes and mugs), can be mainly attributed to an inability to obtain informative features on the object. It is worth noting that in these cases, either the raw Google performance was very good (mugs) or the portion of good images was very small ($\leq 25\%$).

4.2 Unsupervised Learning

In this approach, 6 of the 10 cases were significantly better than the raw Google output. Many of them were only slightly worse than the supervised case, with the motorbike category actually superior. This category is shown in figure 7.

In table 2, RANSAC-style learning is compared to learning directly from all images in the dataset. The proportion of junk images in the dataset determines which of the two approaches is superior: using all images is marginally better when the proportion is small, while the RANSAC approach is decisively better with a large proportion of junk.

5 Discussion and Future Work

Reranking Google images based on their similarity is a problem that is similar to classical visual object recognition. However, it is worth noting the significant differences. In the classical setting of visual recognition we are handed a clean training set consisting of carefully labelled 'positive' and 'negative' examples; we are then asked to test our algorithm on fresh data that was collected independently. In the present scenario the training set is not labelled, it contains a minority (20-50%) of 'good' examples, and a majority of either 'intermediate' or 'junk' examples. Moreover, after learning, our task is to sort the 'training' set, rather than work on fresh data.

Selecting amongst models composed of heterogeneous features is a difficult challenge in our setting. If we had the luxury of a clean labelled training set, then part of this could have been selected as a validation set and then used to select between all-curve and all-patch models. Indeed we could then have trained heterogeneous models where parts could be either curves or patches. However, the non-parametric RPC scoring methods developed here are not up to this task.

It is clear that the current features used are somewhat limited in that they capture only a small fraction of the information from each image. In some of the datasets (e.g. horses) the features did not pick out the distinctive information of the category at all, so the model had no signal to deal with and the algorithm failed as a consequence. By introducing a wider range of feature types (e.g. corners, texture) a wider range of datasets should be accessible to the algorithm.

Overall, we have shown that in the cases where the model's features (patches and curves) are suitable for the object class, then there is a marked improvement in the ranking. Thus we can conclude that the conjecture of the introduction is valid – visual consistency ranking is a viable visual category filter for these datasets.

There are a number of interesting issues in machine learning and machine vision that emerge from our experience: (a) Priors were not used in either of the learning

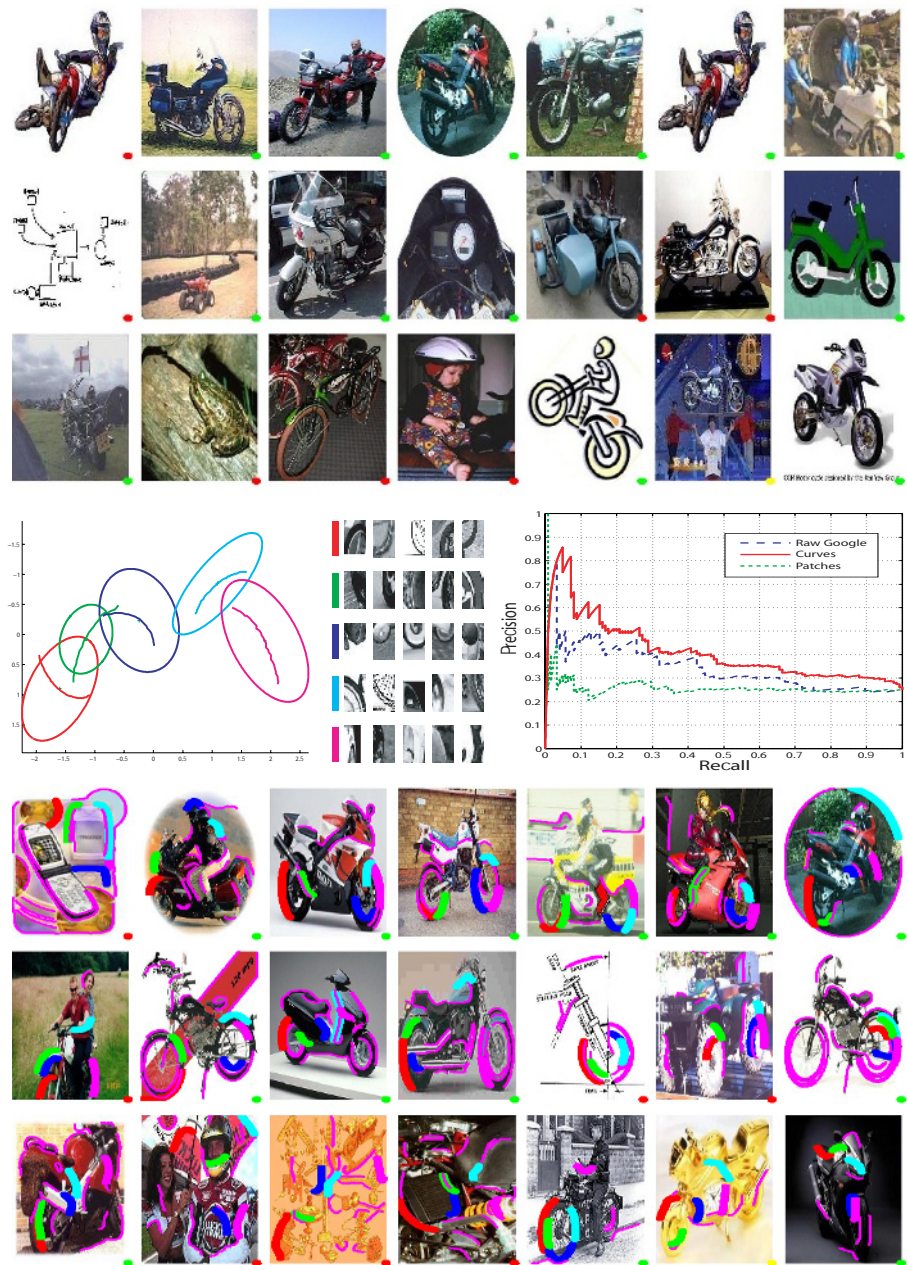


Fig. 7. Motorbike. The top scoring unsupervised motorbike model, selected automatically. The model picks up on the wheels of the bike, despite a wide range of viewpoints and clutter. The RPC (good vs intermediate & junk) shows the curve model performing better than Google’s raw output and the model based on patches (which is actually worse than the raw output).

scenarios. In Fei-Fei *et al.* [8] priors were incorporated into the learning process of the constellation model, enabling effective models to be trained from a few images. Applying these techniques should enhance the performance of our algorithm. (b) The ‘supervised’ case could be improved by using simultaneously the small labelled training data provided by the user, as well as the large unlabelled original dataset. Machine learning researchers are making progress on the problem of learning from ‘partially labeled’ data. We ought to benefit from that effort.

Acknowledgements. Financial support was provided by: EC Project CogViSys; UK EPSRC; Caltech CNSE and the NSF.

References

1. Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11(7):1691–1715, 1999.
2. J. Bach, C. Fuller, R. Humphrey, and R. Jain. The virage image search engine: An open framework for image management. In *SPIE Conf. on Storage and Retrieval for Image and Video Databases*, volume 2670, pages 76–87, 1996.
3. E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *Proc. ECCV*, pages 109–124, 2002.
4. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *7th Int. WWW Conference*, 1998.
5. M. Burl, T. Leung, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *Proc. ECCV*, 1998.
6. J. F. Canny. A computational approach to edge detection. *IEEE PAMI*, 8(6):679–698, 1986.
7. T. Deselaers, D. Keysers, and H. Ney. Clustering visually similar images to improve image search engines. In *Informatiktage 2003 der Gesellschaft für Informatik, Bad Schussenried, Germany.*, 2003.
8. L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, pages 1134–1141, 2003.
9. P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. In *Proc. CVPR*, pages 2066–2073, 2000.
10. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, 2003.
11. M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395, 1981.
12. T. Gevers and A. W. M. Smeulders. Content-based image retrieval by viewpoint-invariant color indexing. *Image and vision computing*, 17:475–488, 1999.
13. B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. Categorization by learning and combining object parts. In *Advances in Neural Information Processing Systems 14, Vancouver, Canada.*, volume 2, pages 1239–1245, 2002.
14. T. Kadir and M. Brady. Scale, saliency and image description. *IJCV*, 45(2):83–105, 2001.
15. B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *Proc. CVPR*, 2003.
16. M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.

17. C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy. Planar object recognition using projective shape representation. *IJCV*, 16(2), 1995.
18. C. Schmid. Constructing models for content-based image retrieval. In *Proc. CVPR*, volume 2, pages 39–45, 2001.
19. S. Tong and E. Chang. Support vector machine active learning for image retrieval. *ACM Multimedia*, 2001.
20. N. Vasconcelos and A. Lippman. Learning from user feedback in image retrieval systems. In *Proc. ECCV*, 2000.
21. R. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey. Technical Report UU-CS-2000-34, Department of Computing Science, Utrecht University, 2000.
22. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, pages 511–518, 2001.
23. M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proc. ECCV*, pages 18–32, 2000.

Scene and Motion Reconstruction from Defocused and Motion-Blurred Images via Anisotropic Diffusion

Paolo Favaro¹, Martin Burger², and Stefano Soatto¹

¹ Computer Science Department, UCLA, Los Angeles, CA 90095, USA,
`{favaro,soatto}@cs.ucla.edu`

² Mathematics Department, UCLA, Los Angeles, CA 90095, USA,
`martinb@math.ucla.edu`

Abstract. We propose a solution to the problem of inferring the depth map, radiance and motion of a scene from a collection of motion-blurred and defocused images. We model motion-blur and defocus as an anisotropic diffusion process, whose initial conditions depend on the radiance and whose diffusion tensor encodes the shape of the scene, the motion field and the optics parameters. We show that this model is well-posed and propose an efficient algorithm to infer the unknowns of the model. Inference is performed by minimizing the discrepancy between the measured blurred images and the ones synthesized via forward diffusion. Since the problem is ill-posed, we also introduce additional Tikhonov regularization terms. The resulting method is fast and robust to noise as shown by experiments with both synthetic and real data.

1 Introduction

We consider the problem of recovering the motion, depth map and radiance of a scene from a collection of defocused and motion-blurred images. Defocus is commonly encountered when using cameras with a finite aperture lens, while motion-blur is common when the imaging system is moving. To the best of our knowledge, we are the first to address the above problem. Typically, this problem is approached by considering images that are affected either by defocus or by motion-blur alone. The first case is divided into two fields of research depending on which object one wants to recover. When we are interested in recovering the radiance from defocused (and possibly downsampled) images, we are solving a *super-resolution* problem [2]. If we are interested in recovering the depth map of the scene (and possibly the radiance), then we are solving the so-called problem of *shape from defocus* [8,12,15,17,19,6]. The second case corresponds to the problem of *motion deblurring*, where one is mainly interested in reconstructing the radiance, which can be thought of as the *unblurred* or *ideal* image, of a scene under the assumptions of Lambertian reflection and uniform illumination [3,4,14]. Motion deblurring is a problem of blind deconvolution [5] or blind image restoration [21], and, therefore, is related to a large body of literature [20].

1.1 Contributions of This Paper

The contribution of this paper is twofold: to link the estimation of the depth map of a scene to the recovery of the radiance and to introduce a simple and computationally efficient imaging model for images that are both defocused and motion-blurred. We model motion-blur via the depth map of the scene and the rigid motion of the camera, which requires at most 6 scalar numbers for 2 images (see section 2.2). This model avoids the artifacts of employing oversimplified motion models (e.g. each point on the image plane moves with the same constant velocity) and yields better estimates than motion models where the motion field is completely unconstrained, due to its lower dimensionality. The second contribution of this paper is the introduction of a novel model for defocused and motion-blurred images in the framework of anisotropic diffusion, in the spirit of [10]. The literature on anisotropic diffusion is quite substantial and, therefore, this work relates also to [13,18,16].

We pose the inference problem as the minimization of the discrepancy between the data and the model (i.e. the final value of the anisotropic diffusion for several different focal settings). The problem is ill-posed, it consists in finding a diffusion tensor and an unknown initial value from final values of parabolic equations. For this sake we introduce Tikhonov-type regularization, which also remedies an unwanted effect with respect to motion-blur, where a local minimum would be attained for zero motion in the absence of suitable regularization (see section 3).

2 A General Model for Defocus and Motion-Blur

2.1 An Imaging Model for Space-Varying Defocus

Images captured with a camera are measurements of energy emitted from the scene. We represent an image with a function $J : \Omega \subset \mathbb{R}^2 \mapsto [0, \infty)$, that maps pixels on the image plane to energy values. We assume that Ω is a bounded domain with piecewise smooth boundary $\partial\Omega$. The intensity of the measured energy depends on the distance of the objects in the scene from the camera and the reflectance properties of their surfaces. We describe the surfaces of the objects with a function $s : \mathbb{R}^2 \mapsto [0, \infty)$, and the reflectance with another function $r : \mathbb{R}^2 \mapsto [0, \infty)$; s assigns a depth value to each pixel coordinate and it is called *depth map*. Similarly, r assigns an energy value to each point on the depth map s and it is called, with an abuse of terminology¹, *radiance*. Furthermore, we

¹ In the context of *radiometry*, the term *radiance* refers to a more complex object that describes energy emitted along a certain direction, per solid angle, per foreshortened area and per time instant. However, in our case, since we do not change vantage point and the size of the optics and the CCD are considerably smaller than the size of the scene, each pixel will collect energy mostly from a single direction, and the change in the solid angle between different pixels is approximately negligible. Hence, a function of the position on the surface of the scene, which is the one we use, suffices to describe the variability of the radiance.

usually know lower and upper bounds $0 < s^{min} < s^{max}$ for the depth map s , which we may incorporate as an additional inequality constraint of the form

$$s^{min} \leq s(x) \leq s^{max}, \quad \forall x \in \Omega. \quad (1)$$

The energy measured by an image J also depends on the optics of the camera. We assume the optics can be characterized by a function $h : \Omega \times \mathbb{R}^2 \mapsto [0, \infty)$, the so-called *point spread function* (PSF), so that an image J can be modeled by

$$J(y) = \int h(y, x) r(x) dx. \quad (2)$$

Although we did not write it explicitly, the PSF h depends on the surface s and the parameters of the optics (see section 3 for more details).

Under the assumption that the PSF is Gaussian and that the surface s is smooth, we can substitute the above model with a PDE whose solution $u : \mathbb{R}^2 \times [0, \infty) \mapsto \mathbb{R}$, $(x, t) \mapsto u(x, t)$, at each time t represents an image with a certain amount of blurring. In formulas, we have that $J(y) = u(y, T)$, where T is related to the amount of blurring of J . We use the following *anisotropic diffusion* equation:

$$\begin{cases} \dot{u}(y, t) = \nabla \cdot (D(y) \nabla u(y, t)) & t > 0 \\ u(y, 0) = r(y) & \forall y \in \Omega \\ D(y) \nabla u(y, t) \cdot n = 0 \end{cases} \quad (3)$$

where $D \doteq \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}$ with $d_{ij} : \mathbb{R}^2 \mapsto \mathbb{R}$ for $i, j = 1, 2$ and $d_{12} \equiv d_{21}$, is called *diffusion tensor*. We assume that $d_{ij} \in C^1(\mathbb{R}^2)$ (i.e. the space of functions with continuous partial derivatives in \mathbb{R}^2) for $i, j = 1, 2$, and² $D(y) \geq 0 \forall y \in \mathbb{R}^2$. The symbol ∇ is the gradient operator $\begin{bmatrix} \frac{\partial}{\partial y_1} & \frac{\partial}{\partial y_2} \end{bmatrix}^T$ with $y = [y_1 \ y_2]^T$, and the symbol $\nabla \cdot$ is the divergence operator $\sum_{i=1}^2 \frac{\partial}{\partial y_i}$. n denotes the unit vector orthogonal to $\partial\Omega$. Notice that there is a scale ambiguity between the time T and the determinant of the diffusion tensor D . We will set $T = \frac{1}{2}$ to resolve this ambiguity.

When the depth map s is a plane parallel to the image plane, the PSF h is a Gaussian with constant covariance σ^2 , and it is easy to show that $2tD = \sigma^2 I_d$, where I_d is the 2×2 identity matrix. In particular, at time $t = T = \frac{1}{2}$ we have $D = \sigma^2 I_d$. This model is fairly standard and was used for instance in [10].

2.2 An Imaging Model for Motion-Blur

On the image plane we measure projections of three dimensional points in the scene. In other words, given a point $X(t) = [X_1(t) \ X_2(t) \ X_3(t)] \in \mathbb{R}^3$ at a time instant t , we measure

$$x(t) \doteq [x_1(t) \ x_2(t)]^T \doteq \begin{bmatrix} \frac{X_1(t)}{X_3(t)} & \frac{X_2(t)}{X_3(t)} \end{bmatrix}^T. \quad (4)$$

² Since D is a tensor, the notation $D(y) \geq 0$ means that $D(y)$ is positive semi-definite.

Using the projections of the points on the image plane $x(t)$, we can write the coordinates of a point $X(t)$ as

$$X(t) = [x(t)^T \ 1]^T s(x(t)). \quad (5)$$

We denote with $V = [V_1(t) \ V_2(t) \ V_3(t)]^T \in \mathbb{R}^3$ the translational velocity and with $\omega \in \mathbb{R}^3$ the rotational velocity of the scene. Then, it is well known that the time derivative of the projection x satisfies (see [11] for more details):

$$\dot{x}(t) = \frac{1}{s(x(t))} \begin{bmatrix} 1 & 0 & -x_1(t) \\ 0 & 1 & -x_2(t) \end{bmatrix} V + \begin{bmatrix} -1 - x_2^2(t) & x_1(t)x_2(t) & -x_2(t) \\ -x_1(t)x_2(t) & 1 + x_1^2(t) & x_1(t) \end{bmatrix} \omega. \quad (6)$$

We define $v \doteq \dot{x}(t)$ and call it the *velocity field*.

As we have anticipated, we restrict ourselves to a crude motion model that only represents translations parallel to the image plane, i.e.

$$v(t) = \frac{V_{1,2}(t)}{s(x(t))} \quad (7)$$

where $V_{1,2}$ is the velocity in focal length units. Now, recalling eq. (2), we have that $J(x+vt)$ denotes an image captured at time t . If the camera shutter remains open while moving the camera with velocity V for a time interval ΔT , then the image I we measure on the image plane can be written as:

$$I(x) = \frac{1}{\Delta T} \int_{-\frac{\Delta T}{2}}^{\frac{\Delta T}{2}} J(x+vt) dt \simeq \int \frac{1}{\sqrt{2\pi\gamma^2}} e^{-\frac{t^2}{2\gamma^2}} J(x+vt) dt \quad (8)$$

where γ depends on the time interval ΔT . The parameter γ can be included in the velocity vector v since there is an ambiguity between the duration of the integration time and the magnitude of the velocity. Therefore, we have

$$I(x) = \int \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} J(x+vt) dt. \quad (9)$$

For simplicity, the above model has been derived for the case of a sideways translational motion, but it is straightforward to extend it to the general case of eq. (6).

2.3 Modeling Motion-Blur and Defocus Simultaneously

In this section, we consider images where defocus and motion-blur occur simultaneously. In the presence of motion, a defocused image J measured at time t can be expressed as

$$J(y+vt) = \int h(y+vt, x) r(x) dx. \quad (10)$$

Following eq. (9), we obtain

$$I(y) = \int \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} \int \frac{1}{2\pi\sigma^2} e^{-\frac{(y-x+vt)^T(y-x+vt)}{2\sigma^2}} r(x) dx dt. \quad (11)$$

If we now interchange the integration order, we can write the previous equation in a more compact way as

$$I(y) = \int \frac{1}{2\pi|C|^{\frac{1}{2}}} e^{-\frac{(y-x)^T C^{-1}(y-x)}{2}} r(x) dx \quad (12)$$

where $C = \sigma^2 I_d + vv^T$.

Eq. (12) is also the solution of the anisotropic diffusion PDE (3) with initial condition the radiance r and diffusion tensor $D = \frac{C}{2t}$. Hence, a model for defocused and motion-blurred images is the following:

$$\begin{cases} \dot{u}(y, t) = \nabla \cdot (D \nabla u(y, t)) & t > 0 \\ u(y, 0) = r(y) & \forall y \in \Omega \\ D \nabla u(y, t) \cdot n = 0 \end{cases} \quad (13)$$

where at time $t = T = \frac{1}{2}$, $D = C = \sigma^2 I_d + vv^T$. Now, it is straightforward to extend the model to the space-varying case, and have that

$$D(y) = \sigma^2(y) I_d + v(y)v(y)^T. \quad (14)$$

In particular, when eq. (7) is satisfied, we have

$$D(y) = \sigma^2(y) I_d + \frac{V_{1,2} V_{1,2}^T}{s^2(y)}. \quad (15)$$

Notice that the diffusion tensor just defined is made of two terms: $\sigma^2(y) I_d$ and $\frac{V_{1,2} V_{1,2}^T}{s^2(y)}$. The first term corresponds to the isotropic component of the tensor, and captures defocus. The second term corresponds to the anisotropic component of the tensor, and it captures motion-blur. Furthermore, since both of the terms are guaranteed to be always positive semi-definite, the tensor eq. (15) is positive semi-definite too. We will use eq. (13) together with eq. (15) as our imaging model in all subsequent sections.

2.4 Well-Posedness of the Diffusion Model

A first step in the mathematical analysis is to verify the well-definedness of the parameter-to-output map $(r, s, V_{1,2}) \mapsto u(\cdot, T)$, which corresponds to a well-posedness result for the degenerate parabolic initial-boundary value problems

$$\begin{cases} \dot{u}(y, t) = \nabla \cdot (D(y) \nabla u(y, t)) & t > 0 \\ u(y, 0) = r(y) \\ D(y) \nabla u(y, t) \cdot n = 0 \end{cases} \quad (16)$$

for diffusion tensors of the form $D(y) = \sigma(y)^2 I_d + \frac{V_{1,2} V_{1,2}^T}{s(x)^2}$. n denotes the unit vector orthogonal to the boundary of Ω . The following theorem guarantees the existence of weak solutions for the direct problem:

Theorem 1. *Let $r \in L^2(\Omega)$ and $s \in H^1(\Omega)$ satisfies (1). Then, there exists a unique weak solution $u \in C(0, T; L^2(\Omega))$ of (16), satisfying*

$$\int_0^T \int_{\Omega} \lambda(y) |\nabla u(y, s)|^2 dy ds \leq \int_{\Omega} r(y)^2 dy, \quad (17)$$

where $\lambda(y) \geq 0$ denotes the minimal eigenvalue of $D(y)$.

Proof. See technical report [9].

3 Estimating Radiance, Depth, and Motion

In section 2.1 we introduced the variance σ^2 of the PSF h to model defocus. The variance σ^2 depends on the depth map via $\sigma^2(x) = \left(\frac{d}{2}\right)^2 \left(1 - p \left(\frac{1}{F} - \frac{1}{s(x)}\right)\right)^2$, where d is the aperture of the camera (in pixel units), p is the distance between the image plane and the lens plane, F is the focal length of the lens and s is the depth map of the scene. We simultaneously collect a number N of defocused and motion-blurred images $\{I_1, \dots, I_N\}$ by changing the parameter $p = \{p_1, \dots, p_N\}$. Notice that the parameters p_i lead to different variances $\sigma_i^2(x)$, which affect the isotropic component of the diffusion tensor D , but not its anisotropic component $\frac{V_{1,2} V_{1,2}^T}{s^2(x)}$. As shown in section 2.3 we can represent an image I_i by taking the solution u_i of eq. (13) at time $t = T = 1/2$ with a diffusion tensor $D_i(x) = \sigma_i^2(x) I_d + \frac{V_{1,2} V_{1,2}^T}{s^2(x)}$, and with initial condition $u_i(y, 0) = r(y) \quad \forall i = 1 \dots N$.

We pose the problem of inferring the radiance r , the depth map s and the motion field v of the scene by minimizing the following least-squares functional with Tikhonov regularization (cf. [7])

$$\begin{aligned} \hat{r}, \hat{s}, \hat{V}_{1,2} = \arg \min_{r, s, V_{1,2}} \sum_{i=1}^N \int_{\Omega} (u_i(x, T) - I_i(x))^2 dx + \alpha \|r - r^*\|^2 + \beta \|\nabla s\|^2 + \\ + \gamma (\|V_{1,2}\| - M)^2, \end{aligned} \quad (18)$$

where α, β , and γ are positive regularization parameters, r^* is a prior³ for r and M is a suitable positive number⁴. One can choose the norm $\|\cdot\|$ depending on the desired space of solutions. We choose the L_2 norm for the radiance and the components of the gradient of the depth map and the ℓ_2 norm for the velocity vector $V_{1,2}$. In this functional, the first term takes into account the discrepancy between the model and the measurements; the second and third term are classical regularization functionals, imposing some regularity on the estimated depth map

³ We do not have a preferred prior for the radiance r . However, it is necessary to introduce this term to guarantee that the estimated radiance does not diverge. In practice, one can use as a prior r^* one of the input images, or a combination of them, and choose a very small α .

⁴ Intuitively, the constant M is related to the maximum degree of motion-blur that we are willing to tolerate in the input data.

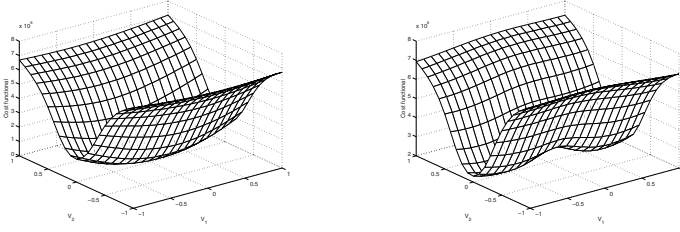


Fig. 1. Left: cost functional for various values of V_1 and V_2 when $\gamma = 0$ or $M = 0$. Right: cost functional for various values of V_1 and V_2 when $\gamma \neq 0$ and $M \neq 0$. In both cases the cost functional eq. (18) is computed for a radiance \hat{r} and a depth map \hat{s} away from the true radiance r and the true depth map s . Notice that on the right plot there are two symmetric minima for $V_{1,2}$. This is always the case unless the true velocity satisfies $V_{1,2} = 0$, since the true $V_{1,2}$ can be determined only up to the sign.

and penalizing large deviations of the radiance from the prior. The last term is of rather unusual form, its main objective being to exclude $V_{1,2} = 0$ as a stationary point. One easily checks that for $\gamma = 0$ or $M = 0$, $V_{1,2} = 0$ is always a stationary point of the functional in (18), which is of course an undesirable effect. This stationary point is removed for positive values of M and γ (see Figure 1).

3.1 Cost Functional Minimization

To minimize the cost functional (18) we employ a gradient descent flow. For each unknown we compute a sequence converging to a local minimum of the cost functional, i.e. we have sequences $\hat{r}(x, \tau)$, $\hat{s}(x, \tau)$, $\hat{V}_{1,2}(\tau)$, such that $\hat{r}(x) = \lim_{\tau \rightarrow \infty} \hat{r}(x, \tau)$, $\hat{s}(x) = \lim_{\tau \rightarrow \infty} \hat{s}(x, \tau)$, $\hat{V}_{1,2} = \lim_{\tau \rightarrow \infty} \hat{V}_{1,2}(\tau)$. At each iteration we update the unknowns by moving in the opposite direction of the gradient of the cost functional with respect to the unknowns. In other words, we let $\partial \hat{r}(x, \tau) / \partial \tau \doteq -\nabla_{\hat{r}} E(x)$, $\partial \hat{s}(x, \tau) / \partial \tau \doteq -\nabla_{\hat{s}} E(x)$, $\partial \hat{V}_{1,2}(\tau) / \partial \tau \doteq -\nabla_{\hat{V}_{1,2}} E(x)$. It can be shown that the above iterations decrease the cost functional as τ increases. The computation of the above gradients is rather involved, but yields the following formulas, that can be easily implemented numerically:

$$\begin{aligned} \nabla_r E &= \sum_{i=1}^N w_i(x, 0) \\ \nabla_s E &= 2 \sum_{i=1}^N \int_0^T \left(\sigma_i(x) \frac{p_i}{s^2(x)} I_d + \frac{V_{1,2} V_{1,2}^T}{s^3(x)} \right) \nabla u_i(x, t) \cdot \nabla w_i(x, t) dt \quad (19) \\ \nabla_{V_{1,2}} E &= - \sum_{i=1}^N \int_0^T \int_{\Omega} \left(\frac{V'_{1,2} V_{1,2}^T + V_{1,2} V_{1,2}^T}{s^2(x)} \nabla u_i(x, t) \cdot \nabla w_i(x, t) \right) dx dt \end{aligned}$$

where w_i satisfies the following adjoint parabolic equation (see [9] for more details):

$$\begin{cases} \dot{w}_i(y, t) = -\nabla \cdot (D_i(y) \nabla w_i(y, t)) \\ w_i(y, T) = u_i(y, T) - I_i(y) \\ (D_i(y) \nabla w_i(y, t)) \cdot n = 0. \end{cases} \quad (20)$$

4 Experiments

The algorithm presented in section 3.1 is tested on both synthetic (section 4.1) and real (section 4.2) data. In the first case, we compare the estimated unknowns with the ground truth and establish the performance of the algorithm for different amounts of noise. In the second case, since we do not have the ground truth, we only present a qualitative analysis of the results. We implement the gradient flow equations in section 3.1 with standard finite difference schemes (see [18,1]).

4.1 Synthetic Data

In this first set of experiments, we consider a scene made of a slanted plane (see the leftmost image in Figure 4), that has one side at $0.52m$ from the camera and the opposite side at $0.85m$ from the camera. The slanted plane is painted with a random texture. We define the radiance r to be the image measured on the image plane when a pinhole lens is used (see first image from the left in Figure 2). The second image from the left in Figure 2 has been captured when the scene or the camera are subject to a translational motion while the camera shutter remains open. Notice that the top portion of the image is subject to a more severe motion-blur than the bottom part. This is due to the fact that in this case points that are far from the camera (bottom portion of the image) move at a slower speed than points that are close to the camera (top portion of the image).

We simulate a camera that has focal length $0.012m$ and F-number 2. With these settings we capture two images: one by focusing at $0.52m$, and the other by focusing at $0.85m$. If neither the camera nor the scene are moving, we capture the two rightmost images shown in Figure 2. Instead, if either the camera or the scene are moving sideways, we capture the two leftmost images shown in Figure 3. The latter two are the images we give in input to our algorithm. In Figure 3 we show the recovered radiance when no motion-blur is taken into account (third image from the left) and when motion-blur is taken into account (rightmost image). As one can notice by visual inspection, the latter estimate of the radiance is sharper than the estimate of the radiance when motion-blur is not modeled. The improvement in the estimation of the radiance can also be evaluated quantitatively since we have ground truth. To measure the accuracy of the estimated radiance, we compute the following normalized RMS error:

$$NRMSE(\phi_{estimated}, \phi_{true}) = \frac{\|\phi_{estimated} - \phi_{true}\|}{\|\phi_{true}\|} \quad (21)$$

where $\phi_{estimated}$ is the estimated unknown, ϕ_{true} is the ground truth and $\|\cdot\|$ denotes the L_2 norm. We obtain that the NRMSE between the true radiance

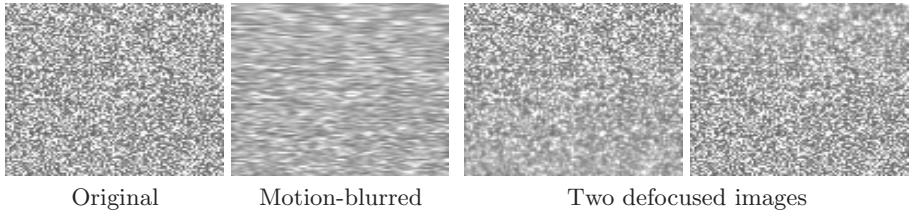


Fig. 2. First from the left: synthetically generated radiance. Second from the left: motion-blurred radiance. This image has been obtained by motion-blurring the synthetic radiance on the left. Third and fourth from the left: defocused images from a scene made of the synthetic radiance in Figure 2 (leftmost) and depth map in Figure 4 (leftmost) without motion-blur.

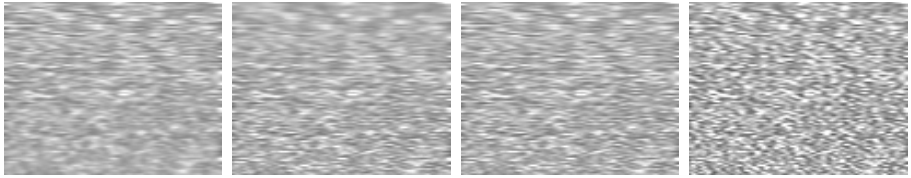


Fig. 3. First and second from the left: defocused and motion-blurred images from a scene made of the synthetic radiance in Figure 2 (leftmost) and depth map in Figure 4 (leftmost). Third from the left: recovered radiance from the two defocused and motion-blurred images on the left when no motion-blur is taken into account ($V_{1,2} = 0$). Fourth from the left: recovered radiance from the two defocused and motion-blurred images on the left when motion blur is taken into account ($V_{1,2} \neq 0$).



Fig. 4. Left: true depth map of the scene. Middle: recovered depth map. Right: profile of the recovered depth map. As can be noticed, the recovered depth map is very close to the true depth map with the exception of the top and bottom sides. This is due to the higher blurring that the images are subject to at these locations.

and the motion-blurred radiance (second image from the left in Figure 2) is 0.2636. When we compensate only for defocus during the reconstruction, the NRMSE between the true radiance and the recovered radiance is 0.2642. As expected, since the motion-blurred radiance is the best estimate possible when we do not compensate for motion-blur, this estimated radiance cannot be more accurate than the motion-blurred radiance. Instead, when we compensate for both defocus and motion-blur, the NRMSE between the true radiance and the recovered radiance is 0.2321. This shows that the outlined algorithm can restore

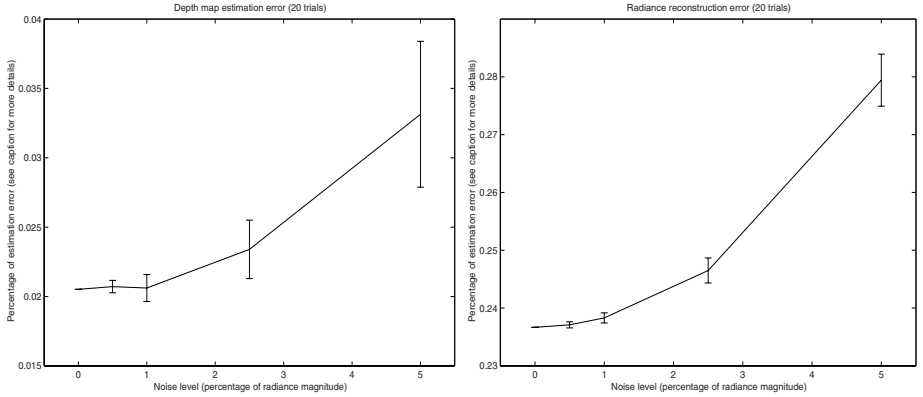


Fig. 5. Left: depth map estimation for 5 levels of additive Gaussian noise. The plot shows the error bar for 20 trials with a staircase depth map and a random radiance. We compute the RMS error between the estimated depth and the true depth, and normalize it with respect to the norm of the true depth (see eq. (21)). Right: radiance estimation for 5 levels of additive Gaussian noise. As in the previous error bar, we compute the RMS error between the true radiance and the reconstructed radiance and then normalize it with respect to the norm of the true radiance.

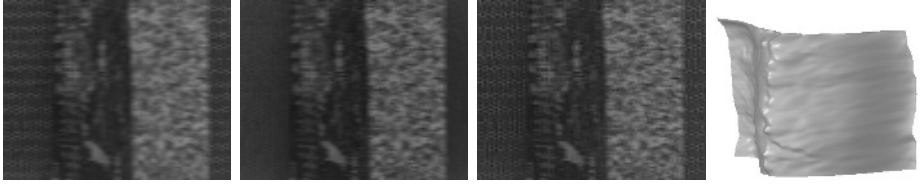


Fig. 6. First and second from the left: input images of the first data set. The two images are both defocused and motion-blurred. Motion-blur is caused by a sideways motion of the camera. Third from the left: recovered radiance. Fourth from the left: recovered depth map.

images that are not only defocused, but also motion-blurred. The recovered depth map is shown in Figure 4 on the two rightmost images together with the ground truth for direct comparison (left). The true motion is $V_{1,2} = [0.8 \ 0]^T$ and the recovered motion is $[0.8079 \ -0.0713]^T$ in focal length units.

To test the performance and the robustness of the algorithm, we synthetically generate defocused and motion-blurred images with additional Gaussian noise. We use a scene made of a staircase depth map with 20 steps, with the first step at $0.52m$ from the camera and the last step at $0.85m$ from the camera. As in the previous experiment, we capture two images: one by focusing at $0.52m$ and the other by focusing at $0.85m$. To each of the images we add the following 5 different amounts of Gaussian noise: 0%, 0.5%, 1%, 2.5% and 5% of the radiance magnitude. For each noise level we run 20 experiments from which we compute the mean and the standard deviation of the NRMSE. The results are shown in Figure 5.

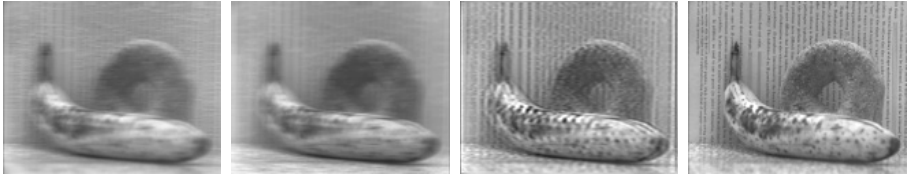


Fig. 7. First and second from the left: input images of the second data set. The two images are both defocused and motion-blurred. Motion-blur is caused by a sideways motion of the camera. Third from the left: recovered radiance. Fourth from the left: an image taken without motion-blur.

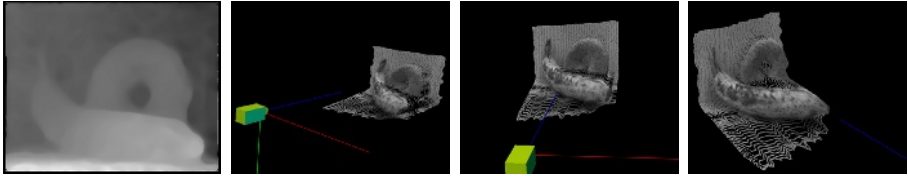


Fig. 8. First from the left: estimated depth map visualized as a gray level intensity image. Second, third and fourth from the left: visualization of the estimated depth map from different viewing angles. The depth map is also texture mapped with the estimated radiance.

4.2 Real Images

We test the algorithm on two data sets. The first data set is made of the two real images shown in Figure 6. The scene is made of a box that is moving sideways. We simultaneously capture two images with a multifocal camera kindly lent to us by S. K. Nayar. The camera has an AF NIKKOR 35mm Nikon lens, with F-number 2.8. We capture the first image by focusing at 70mm from the camera and the second image by focusing at 90mm from the camera. The scene lies entirely between 70mm and 90mm. The estimated radiance is shown in Figure 6, together with the recovered depth map. The estimated motion is $V_{1,2} = [0.5603 \ 0.0101]^T$ in units of focal length. In the second data set we use the two defocused and motion-blurred images in Figure 7 (first and second image from the left) captured with the same camera settings as in the first data set. The scene is composed of a banana and a bagel and the scene is moving sideways. The estimated radiance is shown in the third image from the left of the same figure. To visually compare the quality of the estimated radiance, we also add the fourth image from the left in Figure 7. This image has been obtained from about the same viewing point when neither the camera nor the scene was moving. Hence, this image is only subject to defocus. The reconstructed depth map is shown in Figure 8. The first image from the left is the depth map visualized as a gray level image. Light intensities correspond to points that are close to the camera and dark intensities correspond to points that are far from the camera. The next three images are visualizations of the depth map from different viewing angles with the estimated radiance texture mapped onto it. The estimated velocity for this data set is $V_{1,2} = [0.9639 \ -0.0572]^T$, that corresponds to a sideways motion.

5 Summary and Conclusions

In this manuscript we proposed a solution to the problem of inferring the depth, radiance and motion of a scene from a collection of motion-blurred and defocused images. First, we presented a novel model that can take into account both defocus and motion-blur (assuming motion is pure sideways translation), and showed that it is well-posed. Motion-blurred and defocused images are represented as the solution of an anisotropic diffusion equation, whose initial conditions are defined by the radiance and whose diffusion tensor encodes the shape of the scene, the motion field and the optics parameters. Then, we proposed an efficient algorithm to infer the unknowns of the model. The algorithm is based on minimizing the discrepancy between the measured blurred images and the ones synthesized via diffusion. Since the inverse problem is ill-posed, we also introduce additional Tikhonov regularization terms. The resulting method is fast and robust to noise as shown in the experimental section.

Acknowledgements. This research has been supported by funds AFOSR F49620-03-1-0095, ONR N00014-03-1-0850, NIH PRE-NDEBC, NSF RHA 115-0208197.

References

1. G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing*. Springer-Verlag, 2002.
2. Simon Baker and Takeo Kanade. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, September 2002.
3. B. Basclé, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *European Conference on Computer Vision*, volume 2, pages 573–582, 1996.
4. M. Ben-Ezra and S.K. Nayar. Motion deblurring using hybrid imaging. In *Computer Vision and Pattern Recognition*, volume 1, pages 657–664, 2003.
5. M. Bertero and P. Boccacci. Introduction to inverse problems in imaging. *Institute of Physics Publishing, Bristol and Philadelphia*, 1998.
6. S. Chaudhuri and A. Rajagopalan. *Depth from defocus: a real aperture imaging approach*. Springer Verlag, 1999.
7. H. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer Academic Publishers, Dordrecht, 1996.
8. J. Ens and P. Lawrence. An investigation of methods for determining depth from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:97–108, 1993.
9. P. Favaro, M. Burger, and S. Soatto. Scene and motion reconstruction from defocused and motion-blurred images via anisotropic diffusion. In *UCLA - Math. Dept. Technical Report (cam03-63)*, November 2003.
10. P. Favaro, S. Osher, S. Soatto, and L. Vese. 3d shape from anisotropic diffusion. In *Intl. Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 179–186, 2003.

11. D.J. Heeger and A.D. Jepson. Subspace methods for recovering rigid motion i. *Int. J. of Computer Vision*, 7(2):95–117, 1992.
12. A. Pentland. A new sense for depth of field. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9:523–531, 1987.
13. P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–39, 1990.
14. A. Rav-Acha and S. Peleg. Restoration of multiple images with motion blur in different directions. In *IEEE Workshop on Applications of Computer Vision (WACV)*, Palm-Springs, 2000.
15. M. Subbarao and G. Surya. Depth from defocus: a spatial domain approach. *Intl. J. of Computer Vision*, 13:271–294, 1994.
16. D. Tschumperle and R. Deriche. Vector-valued image regularization with pde's: A common framework for different applications. In *CVPR03*, pages I: 651–656, 2003.
17. M. Watanabe and S. Nayar. Rational filters for passive depth from defocus. *Intl. J. of Comp. Vision*, 27(3):203–225, 1998.
18. J. Weickert. *Anisotropic Diffusion in Image Processing*. B.G.Teubner Stuttgart, 1998.
19. Y. Xiong and S. Shafer. Depth from focusing and defocusing. In *Proc. of the Intl. Conf. of Comp. Vision and Pat. Recogn.*, pages 68–73, 1993.
20. Y. Yitzhaky, R. Milberg, S. Yohaev, and N. S. Kopeika. Comparison of direct blind deconvolution methods for motion-blurred images. In *Applied Optics-IP*, volume 38, pages 4325–32, July 1999.
21. Y. You and M. Kaveh. Blind image restoration by anisotropic diffusion. *IEEE Trans. on Image Processing*, 8(3):396–407, 1999.

Semantics Discovery for Image Indexing

Joo-Hwee Lim¹ and Jesse S. Jin²

¹ Institute for Infocomm Research

21 Heng Mui Keng Terrace, Singapore 119613

jooohwee@i2r.a-star.edu.sg

² University of New South Wales, Sydney 2052, Australia

jesse@cse.unsw.edu.au

Abstract. To bridge the gap between low-level features and high-level semantic queries in image retrieval, detecting meaningful visual entities (e.g. faces, sky, foliage, buildings etc) based on trained pattern classifiers has become an active research trend. However, a drawback of the supervised learning approach is the human effort to provide labeled regions as training samples. In this paper, we propose a new three-stage hybrid framework to discover local semantic patterns and generate their samples for training with minimal human intervention. Support vector machines (SVM) are first trained on local image blocks from a small number of images labeled as several semantic categories. Then to bootstrap the local semantics, image blocks that produce high SVM outputs are grouped into Discovered Semantic Regions (DSRs) using fuzzy c-means clustering. The training samples for these DSRs are automatically induced from cluster memberships and subject to support vector machine learning to form local semantic detectors for DSRs. An image is then indexed as a tessellation of DSR histograms and matched using histogram intersection. We evaluate our method against the linear fusion of color and texture features using 16 semantic queries on 2400 heterogeneous consumer photos. The DSR models achieved a promising 26% improvement in average precision over that of the feature fusion approach.

1 Introduction

Content-based image retrieval research has progressed from the feature-based approach (e.g. [9]) to the region-based approach (e.g. [5]). In order to bridge the semantic gap [20] that exists between computed perceptual visual features and conceptual user query expectation, detecting semantic objects (e.g. faces, sky, foliage, buildings etc) based on trained pattern classifiers has received serious attention (e.g. [15,16,22]). However, a major drawback of the supervised learning approach is the human effort required to provide labeled training samples, especially at the image region level. Lately there are two promising trends that attempt to achieve semantic indexing of images with minimal or no effort of manual annotation (i.e. semi-supervised or unsupervised learning).

In the field of computer vision, researchers have developed object recognition systems from unlabeled and unsegmented images [8,19,25]. In the context

of relevance feedback, unlabeled images have also been used to bootstrap the learning from very limited labeled examples (e.g. [24,26]). For the purpose of image retrieval, unsupervised models based on “generic” texture-like descriptors without explicit object semantics can also be learned from images without manual extraction of objects or features [18]. As a representative of the state-of-the-art, sophisticated generative and probabilistic model has been proposed to represent, learn, and detect object parts, locations, scales, and appearances from fairly cluttered scenes with promising results [8].

Motivated from a machine translation perspective, object recognition is posed as a lexicon learning problem to translate image regions to corresponding words [7]. More generally, the joint distribution of meaningful text descriptions and entire or local image contents are learned from images or categories of images labeled with a few words [1,3,11,12]. The lexicon learning metaphor offers a new way of looking at object recognition [7] and a powerful means to annotate entire images with concepts evoked by what is visible in the image and specific words (e.g. fitness, holiday, Paris etc [12]). While the annotation results on entire images look promising [12], the correspondence problem of associating words with segmented image regions remains very challenging [3] as segmentation, feature selection, and shape representation are critical and non-trivial choices [2].

In this paper, we address the issue of minimal supervision differently. We do not assume availability of text descriptions for image or image classes as in [3, 12]. Neither do we know the object classes to be recognized as in [8]. We wish to discover and associate local unsegmented regions with semantics and generate their samples to construct models for content-based image retrieval, all with minimal manual intervention. This is realized as a novel three-stage hybrid framework that interleave supervised and unsupervised learnings. First support vector machines (SVM) are trained on local image blocks from a small number of images labeled as several semantic categories. Then to bootstrap the local semantics, *typical* image blocks that produce high SVM outputs are grouped into Discovered Semantic Regions (DSRs) using fuzzy c-means clustering. The training samples for these DSRs are automatically induced from cluster memberships and subject to local support vector machine learning to form local semantic detectors for DSRs. An image is indexed as a tessellation of DSR histograms and matched using histogram intersection.

We evaluate our method against the linear fusion of color and texture features using 16 semantic queries on 2400 heterogeneous consumer photos with many cluttered scenes. The DSR implementation achieved a promising 26% improvement in average precision over that of the feature fusion approach.

The rest of the paper is presented as follows. We explain our local semantics discovery framework followed by the mechanisms for image indexing and matching in the next two sections respectively. Then we report and compare the results on the query-by-example experiments. Last but not least, we discuss the relevant aspects of our approach with other promising works in unsupervised semantics learning and issues for future research.

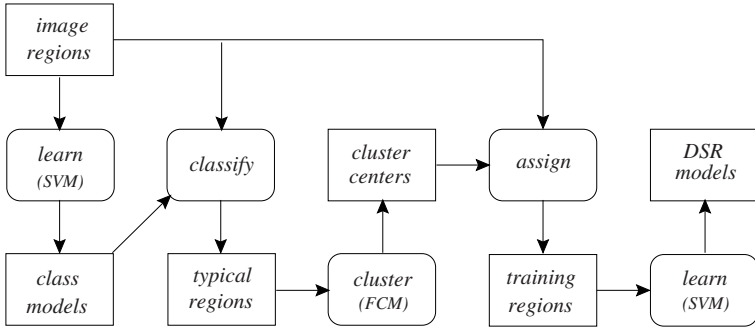


Fig. 1. A schematic diagram of local semantics discovery

2 Local Semantics Discovery

Image categorization is a powerful divide-and-conquer metaphor to organize and access images. Once the images are sorted into semantic classes, searching and browsing can be carried out in more effective and efficient way by focusing only at relevant classes and subclasses. Moreover the classes provide context for other tasks. For example, for medical images, the context could be the pathological classes for diagnostic purpose [4] or imaging modalities for visualization purpose [14]. In this paper, we propose a framework to discover the local semantics that distinguish image classes and use these Discovered Semantic Regions (DSRs) to span a semantic space for image indexing. Fig. 1 depicts the steps in the framework which can be divided into three learning phases as described below.

2.1 Learning of Local Class Semantics

Given a content or application domain, some distinctive classes C_k with their image samples are identified. For consumer images used in our experiments, a taxonomy as shown in Fig. 2 has been designed. This hierarchy of 11 categories is more comprehensive than the 8 categories addressed in [23]. We select the 7 disjoint categories represented by the leaf nodes (except the **miscellaneous** category) in Fig. 2 and their samples to train 7 binary support vector machines (SVM). The training samples are tessellated image blocks z from the class samples. After learning, the class models would have captured the local class semantics and a high SVM output (i.e. $C_k(z) \gg 0$) would suggest that the local region z is typical to the semantics of class k .

In this paper, as our test data are heterogeneous consumer photos, we extract color and textures features for a local image block and denote this feature vector as z . Hence a feature vector z has two parts, namely, a color feature vector z^c and a texture feature vector z^t . For the color feature, as the image patch for training and detection is relatively small, the mean and standard deviation of each color channel is deemed sufficient (i.e. z^c has 6 dimensions). In our experiments, we use the YIQ color space over other color spaces (e.g. RGB,

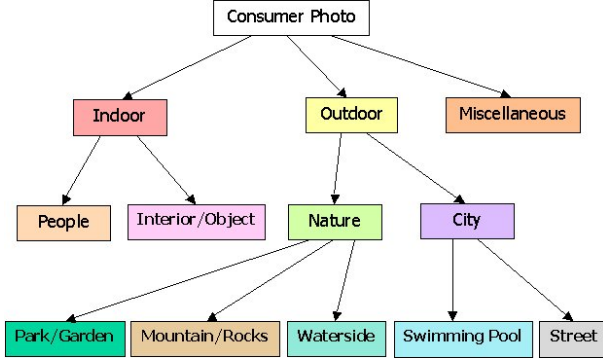


Fig. 2. Proposed hierarchy of consumer photo categories

HSV, LUV) as it performed better in our experiments. For the texture feature, we adopted the Gabor coefficients which have been shown to provide excellent pattern retrieval results [13]. Similarly, the means and standard deviations of the Gabor coefficients (5 scales and 6 orientations) in an image block are computed as z^t which has 60 dimensions. To normalize both the color and texture features, we use the Gaussian (i.e. zero-mean) normalization.

The distance or similarity measure depends on the kernel adopted for the support vector machines. For the experimental results reported in this paper, we adopted polynomial kernels with the following modified dot product similarity measure between feature vectors y and z ,

$$y \cdot z = \frac{1}{2} \left(\frac{y^c \cdot z^c}{|y^c||z^c|} + \frac{y^t \cdot z^t}{|y^t||z^t|} \right) \quad (1)$$

2.2 Learning of Typical Semantic Partitions

With the help of the learned class models C_k , we can generate sets of local image regions that characterize the class semantics (which in turn captures the semantic of the content domain) \mathcal{X}_k as

$$\mathcal{X}_k = \{z | C_k(z) > \rho\} \quad (\rho \geq 0) \quad (2)$$

However, the local semantics hidden in each \mathcal{X}_k is opaque and possibly multi-mode. We would like to discover the multiple groupings in each class by unsupervised learning such as Gaussian mixture modeling and fuzzy c-means clustering. The result of the clustering is a collection of partitions m_{kj} , $j = 1, 2, \dots, N_k$ in the space of local semantics for each class, where m_{kj} are usually represented as cluster centers and N_k are the numbers of partitions for each class.

2.3 Learning of Discovered Semantic Regions

After obtaining the typical semantic partitions for each class, we can learn the models of DSRs S_i $i = 1, 2, \dots, N$ where $N = \sum_k N_k$ (i.e. linearize m_{kj} subscript

as m_i). We label a local image block ($x \in \cup_k \mathcal{X}_k$) as positive example for S_i if it is closest to m_i and as negative example for S_j $j \neq i$,

$$X_i^+ = \{x | i = \arg \min_t |x - m_t|\} \quad (3)$$

$$X_i^- = \{x | i \neq \arg \min_t |x - m_t|\} \quad (4)$$

where $|\cdot|$ is some distance measure. Now we can perform supervised learning again on X_i^+ and X_i^- using say support vector machines $\mathcal{S}_i(x)$ as DSR models.

To visualize a DSR S_i , we can display the image block s_i that is most typical among those assigned to cluster m_i that belonged to class k ,

$$\mathcal{C}_k(s_i) = \max_{x \in X_i^+} \mathcal{C}_k(x) \quad (5)$$

3 Image Indexing and Matching

Image indexing based on DSRs consists of three steps, namely detection, reconciliation, and aggregation. Once the support vector machines \mathcal{S}_i have been trained, the detection vector T of a local image block z can be computed via the softmax function [6] as

$$T_i(z) = \frac{\exp^{\mathcal{S}_i(z)}}{\sum_j \exp^{\mathcal{S}_j(z)}}. \quad (6)$$

As each binary SVM is regarded as an expert on a DSR, the outputs of \mathcal{S}_i $\forall i$ is set to 0 if there exist some \mathcal{S}_j , $j \neq i$ has a positive output. That is, T_j is close to 1 and $T_i = 0$ $\forall i \neq j$.

To detect DSRs with translation and scale invariance in an image, the image is scanned with multi-scale windows, following the strategy in view-based object detection [17]. In our experiments, we progressively increase the window size from 20×20 to 60×60 at a step of 10 pixels, on a 240×360 size-normalized image. That is, after this detection step, we have 5 maps of detection.

To reconcile the detection maps across different resolutions onto a common basis, we adopt the following principle: If the detection value of the most confident class of a region at resolution r is less than that of a larger region (at resolution $r + 1$) that subsumes the region, then the detection vector of the region should be replaced by that of the larger region at resolution $r + 1$. Using this principle, we start the reconciliation from detection map based on largest scan window (60×60) to detection map based on next-to-smallest scan window (30×30). After 4 cycles of reconciliation, the detection map that is based on the smallest scan window (20×20) would have consolidated the detection decisions obtained at other resolutions.

Suppose a region Z comprises of n small equal regions with feature vectors z_1, z_2, \dots, z_n respectively. To account for the size of detected DSRs in the area Z , the DSR detection vectors of the reconciled detection map are aggregated as

$$T_i(Z) = \frac{1}{n} \sum_k T_i(z_k). \quad (7)$$

For query by examples, the content-based similarity λ between a query q and an image x can be computed in terms of the similarity between their corresponding local regions. For example, the similarity based on L_1 distance measure (city block distance) between query q with m local regions Y_j and image x with m local regions Z_j is defined as

$$\lambda(q, x) = 1 - \frac{1}{2m} \sum_j \sum_i |T_i(Y_j) - T_i(Z_j)| \quad (8)$$

This is equivalent to histogram intersection [21] except that the bins have semantic interpretation. In general, we can attach different weights to the regions (i.e. Y_j, Z_j) to emphasize the focus of attention (e.g. center). In this paper, we report experimental results based on even weights as grid tessellation is used. Also we have attempted various similarity and distance measures (e.g. cosine similarity, L_2 distance, Kullback-Leibler (KL) distance etc) and the simple city block distance in Equation (8) has the best performance. When a query has multiple examples, $Q = \{q_1, q_2, \dots, q_K\}$, the similarity is computed as

$$\lambda(Q, x) = \max_i \lambda(q_i, x) \quad (9)$$

4 Experimental Results

In this paper, we evaluate our DSR-based image indexing approach on 2400 genuine consumer photos, taken over 5 years in several countries with both indoor and outdoor settings. After removing possibly noisy margins, the images are size-normalized to 240×360 . The indexing process automatically detects the layout and applies the corresponding tessellation template. In our experiments, the tessellation for detection of DSRs is a 4×4 grid of rectangular regions. Fig. 3 displays typical photos in this collection. Photos of bad quality (e.g. faded, over-exposed, blurred, dark etc) (not shown here) are retained in order to reflect the complexity of the original data.



Fig. 3. Sample consumer photos from the 2400 collection. They also represent 2 relevant images (top-down, left-right) for each of the 16 queries used in our experiments.

Table 1. Training statistics of the semantic classes C_k for bootstrapping local semantics. The columns (left to right) list the class labels, the size of ground truth, the number of training images, the number of support vectors learned, the number of typical image blocks subject to clustering ($C_k(z) > 2$), and the number of clusters assigned.

Class	G.T.	#trg	#SV	#data	#clus
inob	134	15	1905	1429	4
inpp	840	20	2249	936	5
mtrk	67	10	1090	1550	2
park	304	15	955	728	4
pool	52	10	1138	1357	2
strt	645	20	2424	735	5
wtspd	150	15	2454	732	4



Fig. 4. Most typical image blocks of the DSRs learned (left to right): china utensils and cupboard top (first four) for the **inob** class; faces with different background and body close-up (next five) for the **inpp** class; rocky textures (next two) for the **mtrk** class; green foliage and flowers (next four) for the **park** class; pool side and water (next two) for the **pool** class; roof top, building structures, and roadside (next five) for the **strt** class; and beach, river, pond, far mountain (next four) for the **wtspd** class.

We trained 7 SVMs with polynomial kernels (degree 2, $C = 100$ [10]) for the leaf-node categories (except **miscellaneous**) on color and texture features (Equation (1)) of 60×60 image blocks (tessellated with 20 pixels in both directions) from 105 sample images. Hence each SVM was trained on 16,800 image blocks. After training, the samples from each class k is fed into classifier C_k to test their typicalities. Those samples with SVM output $C_k(z) > 2$ (Equation (2)) are subject to fuzzy c-means clustering. The number of clusters assigned to each class is roughly proportional to the number of training images in each class. Table 1 lists training statistics for these semantic classes: **inob** (indoor interior/objects), **inpp** (indoor people), **mtrk** (mountain/rocks), **park** (park/garden), **pool** (swimming pool), **strt** (street), and **wtspd** (waterside). We have 26 DSRs in total.

To build the DSR models, we trained 26 binary SVM with polynomial kernels (degree 2, $C = 100$ [10]), each on 7467 positive and negative examples (Equations (3) and (4)) (i.e. sum of column 5 of Table 1). To visualize the 26 DSRs that have been learned, we compute the most typical image block for each cluster (Equation (5)) and concatenate their appearances in Fig. 4. Image indexing was based on the steps as explained in Section 3.

Table 2. Results of QBE experiments for 16 semantic queries (left to right): query id, query description, size of ground truth, average precisions based on random retrieval (RAND), linear fusion of color and texture features (CTO), and discovered semantic regions (DSRs) (the indexing for the last two methods are based on 4×4 grid).

Query	Description	G.T.	RAND	CTO	DSR
Q01	indoor	994	0.41	0.62	0.79
Q02	outdoor	1218	0.51	0.78	0.78
Q03	people close-up	277	0.12	0.16	0.33
Q04	people indoor	840	0.35	0.59	0.76
Q05	interior or object	134	0.06	0.18	0.32
Q06	city scene	697	0.29	0.49	0.59
Q07	nature scene	521	0.22	0.35	0.46
Q08	at a swimming pool	52	0.02	0.18	0.62
Q09	street or roadside	645	0.27	0.50	0.53
Q10	along waterside	150	0.06	0.17	0.32
Q11	in a park or garden	304	0.13	0.71	0.51
Q12	at mountain area	67	0.03	0.28	0.31
Q13	buildings close-up	239	0.10	0.35	0.30
Q14	close up, indoor	73	0.03	0.15	0.30
Q15	small group, indoor	491	0.20	0.32	0.45
Q16	large group, indoor	45	0.02	0.29	0.29

We defined 16 semantic queries and their ground truths (G.T.) among the 2400 photos (Table 2). In fact, Fig. 3 shows, in top-down left-to-right order, 2 relevant images for queries Q01-Q16 respectively. As we can see from these sample images, the relevant images for any query considered here exhibit highly varied and complex visual appearance. There is usually no dominant homogeneous color or texture region and they pose great difficulty for image segmentation. Hence to represent each query, we selected 3 (i.e. $K = 3$ in Equation (9)) relevant photos as query examples for Query By Example (QBE) experiments since a single query image is far from satisfactory to capture the semantic of any query and single query images have indeed resulted in poor precisions and recalls in our initial experiments. The precisions and recalls were computed without the query images themselves in the lists of retrieved images.

In our experiments, we compare our local semantic discovery approach (denoted as “DSR”) with the feature-based approach that combines color and texture in a linearly optimal way (denoted as “CTO”). All indexing are carried out with a 4×4 grid on the images.

For the color-based signature, color histograms of b^3 ($b = 4, 5, \dots, 17$) number of bins in the RGB color space were computed on an image. The performance peaked at 2197 ($b = 13$) bins with average precision (over all recall points) $P_{avg} = 0.38$. Histogram intersection [21] was used to compare two color histograms. For the texture-based signature, we adopted the means and standard deviations of Gabor coefficients and the associated distance measure as reported in [13]. The Gabor coefficients were computed with 5 scales and 6 orientations. Convolution windows of $20 \times 20, 30 \times 30, \dots, 60 \times 60$ were attempted. The best performance

Table 3. Comparison of average precisions at top numbers of retrieved images. The last row compares the precisions averaged over all 16 queries. The last column shows the relative improvement in percentage.

Avg.Prec.	CTO	DSR	%
At 20	0.64	0.71	10
At 30	0.59	0.68	15
At 50	0.52	0.63	21
At 100	0.46	0.57	24
<i>overall</i>	<i>0.38</i>	<i>0.48</i>	<i>26</i>

was obtained when 20×20 windows were used with $P_{avg} = 0.24$. The distance measures between a query and an image for the color and texture methods were normalized within $[0, 1]$ and combined linearly. Among the relative weights attempted at 0.1 intervals, the best fusion was obtained at $P_{avg} = 0.38$ with a dominant influence of 0.9 from the color feature.

As shown in Table 2, the DSR approach outperformed or matched the average precisions of the CTO method in all queries except Q11 and Q13. The random retrieval method (i.e. $G.T./2400$) (denoted as “RAND”) was used as a baseline comparison. In particular, the DSR approach more than doubled the performance of RAND and surpassed the average precisions of CTO by at least 0.1 in more than half of the queries (Q03-08, Q10, Q14-15). Averaged over all queries, the DSR approach achieved a 26% improvement in precision over that of CTO (Table 3). As depicted in the same table, DSR is also consistently better than CTO in returning more relevant images at top numbers of images for practical applications. As an illustration, Fig. 5 and Fig. 6 show the query examples and top 18 retrieved images for query Q08 respectively. All retrieved images except image 18 are considered relevant.



Fig. 5. Query images for Q08.



Fig. 6. Top 18 retrieved images by DSR for query Q08.

5 Discussion

For the current implementation of our DSR approach, there are still several issues to be addressed. We can improve the sampling of image blocks for semantic class learning by randomly selecting say 20% of the ground truth images in each class as positive samples (and as negative samples for all other classes) as well as by tessellating image blocks with different sizes (e.g. $20 \times 20, 30 \times 30$ etc) and displacements (e.g. 10 pixels) to generate a more complete and denser coverage of the local semantic space. But these attempts turned out to be too ambitious for practical training.

Another doubt is the usefulness of the semantic class learning in the first place. Can we perform clustering of image blocks in each class directly (i.e. without worrying about $\mathcal{C}_k(z) > \rho$)? The result was indeed inferior (with average precision of 0.39) for the QBE experiments. Hence the typicality criterion is important to pick up the relevant hidden local semantics for discovery.

Cluster validity is a tricky issue. We have tried fixed number of clusters (e.g. 3, 4, 5, 7) and retained large clusters as DSRs. Alternatively we relied on human inspection to select perceptually distinctive clusters (as visualized using Equation (5)) as DSRs. However the current way of assigning number of clusters roughly proportional to the number of training images has produced the best performance in our experiments. In future, we would explore other ways to model DSRs (e.g. Gaussian mixture) and to determine the value of ρ . We would also like to verify our approach on other content domains such as art images, medical images etc to see if the DSRs make sense to the domain experts.

Although our attempt to alleviate the supervised learning requirement of labeled images and regions differs from the current trends of unsupervised object recognition and matching words with pictures, the methods do share some common techniques. For instance, similar to those of Schmid [18] and Fergus et al. [8], our approach computes local region features based on tessellation instead of segmentation though [8] used an interest detector and kept the number of features below 30 for practical implementation. While Schmid focused on “Gabor-like” features [18] and Fergus et al. worked on monochrome information only [8], we have incorporated both color and texture information. As the clusters in [18] were generated by unsupervised learning only, they may not correspond to well-perceived semantics when compared to our DSRs. As we are dealing with cluttered and heterogeneous scenes, we did not model object parts as in the comprehensive case of [8]. On the other hand, we handle scale invariance with multi-scale detection and reconciliation of DSRs during image indexing. Last but not least, while the generative and probabilistic approaches [8,12] may enjoy modularity and scalability in learning, they do not exploit inter-class discrimination to compute features unique to classes as in our case.

For the purpose of image retrieval, the images signatures based on DSRs realize semantic abstraction via prior learning and detection of visual classes when compared to direct indexing based on low-level features. The compact representation that accommodates imperfection and uncertainty in detection also resulted in better performance than the fusion of very high dimension of color

and texture features in our query-by-example experiments. Hence we feel that the computational resources devoted to prior learning of DSRs and their detection during indexing are good trade-off for concise semantic representation and effective retrieval performance. Moreover, the small footprint of DSR signatures has an added advantage in storage space and retrieval efficiency.

6 Conclusion

In this paper, we have presented a hybrid framework that interleaves supervised and unsupervised learning to discover local semantic regions without image segmentation and with minimal human effort. The discovered semantic regions serve as new semantic axes for image indexing and matching. Experimental query-by-example results on 2400 genuine consumer photos with cluttered scenes have shown that images indexes based on the discovered local semantics are more compact and effective over linear combination of color and texture features.

References

1. Barnard, K., Forsyth, D.: Learning the semantics of words and pictures. *Proc. of ICCV* (2001) 408–415
2. Barnard, K., et al.: The effects of segmentation of feature choices in a translation model of object recognition. *Proc. of CVPR* (2003)
3. Barnard, K., et al.: Matching words and pictures. *J. Machine Learning Research* **3** (2003) 1107–1135
4. Brodley, C.E., et al.: Content-based retrieval from medical image databases: A synergy of human interaction, machine learning and computer vision. *Proc. of AAAI* (1999) 760–767
5. Carson, C., et al.: Blobworld: image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. on PAMI* **24** (2002) 1026–1038
6. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford (1995)
7. Duygulu, P., et al.: Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. *Proc. of ECCV* (2002) 97–112
8. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. *Proc. of IEEE CVPR* (2003)
9. Flickner, M., et al.: Query by image and video content: the QBIC system. *IEEE Computer* **28** (1995) 23–30
10. Joachims, T.: Making large-scale SVM learning practical. *Advances in Kernel Methods - Support Vector Learning*. B. Scholkopf, C. Burges, and A. Smola (ed.). MIT-Press (1999)
11. Kutics, A., et al.: Linking images and keywords for semantics-based image retrieval. *Proc. of ICME* (2003) 777–780
12. Li, J., Wang, J.Z.: Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. on PAMI* **25** (2003) 1–14
13. Manjunath, B.S., Ma, W.Y.: Texture features for browsing and retrieval of image data. *IEEE Trans. on PAMI* **18** (1996) 837–842
14. Mojsilovic, A., Gomes, J.: Semantic based categorization, browsing and retrieval in medical image databases. *Proc. of IEEE ICIP* (2002)

15. Naphade, M.R., Kozintsev, I.V., Huang, T.S.: A factor graph framework for semantic video indexing. *IEEE Trans. on CSVT* **12** (2002) 40–52
16. Naphade, M.R., et al.: A framework for moderate vocabulary semantic visual concept detection. *Proc. IEEE ICME* (2003) 437–440
17. Papageorgiou, P.C., Oren, M., Poggio, T.: A general framework for object detection. *Proc. of ICCV* (1997) 555–562
18. Schmid, C.: Constructing models for content-based image retrieval. *Proc. of CVPR* (2001) 39–45
19. Selinger, A., Nelson, R.C.: Minimally supervised acquisition of 3D recognition models from cluttered images. *Proc. of CVPR* (2001) 213–220
20. Smeulders, A.W.M., et al.: Content-based image retrieval at the end of the early years. *IEEE Trans. on PAMI* **22** (2000) 1349–1380
21. Swain, M.J., Ballard, D.N.: Color indexing. *Intl. J. Computer Vision* **7** (1991) 11–32
22. Town, C., Sinclair, D.: Content-based image retrieval using semantic visual categories. Technical Report 2000.14. AT&T Research Cambridge (2000)
23. Vailaya, A., et al.: Bayesian framework for hierarchical semantic classification of vacation images. *IEEE Trans. on Image Processing* **10** (2001) 117–130
24. Wang, L., Chan, K.L., Zhang, Z.: Bootstrapping SVM active learning by incorporating unlabelled images for image retrieval. *Proc. of IEEE CVPR* (2003)
25. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for recognition. *Proc. of ECCV* (2000) 18–32
26. Wu, Y., Tian, Q., Huang, T.S.: Discriminant-EM algorithm with application to image retrieval. *Proc. of CVPR* (2000) 1222–1227

Hand Gesture Recognition within a Linguistics-Based Framework

Konstantinos G. Derpanis, Richard P. Wildes, and John K. Tsotsos

York University, Department of Computer Science and
Centre for Vision Research (CVR)
Toronto Ont. M3J 1P3, Canada
{kosta,wildes,tsotsos}@cs.yorku.ca
<http://www.cs.yorku.ca/~{kosta,wildes,tsotsos}>

Abstract. An approach to recognizing hand gestures from a monocular temporal sequence of images is presented. Of particular concern is the representation and recognition of hand movements that are used in single handed American Sign Language (ASL). The approach exploits previous linguistic analysis of manual languages that decompose dynamic gestures into their static and dynamic components. The first level of decomposition is in terms of three sets of primitives, hand shape, location and movement. Further levels of decomposition involve the lexical and sentence levels and are part of our plan for future work. We propose and demonstrate that given a monocular gesture sequence, kinematic features can be recovered from the apparent motion that provide distinctive signatures for 14 primitive movements of ASL. The approach has been implemented in software and evaluated on a database of 592 gesture sequences with an overall recognition rate of 86.00% for fully automated processing and 97.13% for manually initialized processing.

1 Introduction

Interest in automated gesture recognition has the potential to create powerful human computer interfaces. Computer vision provides methods to acquire and interpret gesture information while being minimally obtrusive to the participant. To be useful, methods must be accurate in recognition with rapid execution to support natural interaction. Further, scalability to encompass the large range of human gestures is important. The current paper presents an approach to recognizing human gestures that leverages both linguistic theory and computer vision methods. Following a path taken in the speech recognition community for the interpretation of speech [22], we appeal to linguistics to define a finite set of contrastive primitives, termed phonemes, that can be combined to represent an arbitrary number of gestures. This ensures that the developed approach is scalable. Currently, we are focused on the representation and recovery of the movement primitives derived from American Sign Language (ASL). This same linguistics analysis has also been applied to other hand gesture languages (e.g. French Sign Language). To affect the recovery of these primitives, we make use

of robust, parametric motion estimation techniques to extract signatures that uniquely identify each movement from a monocular input video sequence. Here, it is interesting to note that human observers are capable of recovering the primitive movements of ASL based on motion information alone [21]. For our case, empirical evaluation suggests that algorithmic instantiation of these ideas has sufficient accuracy to distinguish the target set of ASL movement primitives, with modest processing power.

1.1 Related Research

Significant effort in computer vision has been marshalled in the investigation of human gesture recognition (see [1,20] for general reviews); some examples follow. State-space models have been used to capture the sequential nature of gestures by requiring that a series of states estimated from visual data must match in sequence, to a learned model of ordered states [7]. This general approach also has been used in conjunction with parametric curvilinear models of motion trajectories [6]. An alternative approach has used statistical factored sampling in conjunction with a model of parameterized gestures for recognition [5]; this approach can be seen as an application and extension of the CONDENSATION approach to visual tracking [14]. Further, several approaches have used Hidden Markov Models (HMMs) [17,24,26], neural networks [10] or time-delay neural networks [31] to learn from training examples (e.g., based on 2D or 3D features extracted from raw data) and subsequently recognize gestures in novel input.

A number of the cited approaches have achieved interesting recognition rates, albeit often with limited vocabularies. Interestingly, many of these approaches analyze gestures without breaking them into their constituent primitives, which could be used as in our approach, to represent a large vocabulary from a small set of generative elements. Instead, gestures are dealt with as wholes, with parameters learned from training sets. This tack may limit the ability of such approaches to generalize to large vocabularies as the training task becomes inordinately difficult. Additionally, several of these approaches make use of special purpose devices (e.g., coloured markers, data gloves) to assist in data acquisition.

In [2,28], two of the earliest efforts of using linguistic concepts for the description and recognition of both general and domain specific motion are presented. Recently, at least two lines of investigations have appealed to linguistic theory as an attack on issues in scaling gesture recognition to sizable vocabularies [18, 30]. In [18] the authors use data glove output as the input to their system. Each phoneme, from the parameters shape, location, orientation and movement, is modelled by an HMM based on features extracted from the input stream, with an 80.4% sentence accuracy rate. In [30] to affect recovery, 3D motion is extracted from the scene by fitting a 3D model of an arm with the aid of three cameras in an orthogonal configuration (or a magnetic tracking system). The motion is then fed into parallel HMMs representing the individual phonemes. The authors report that by modelling gestures by phonemes, the word recognition rate was not severely diminished, 91.19% word accuracy with phonemes

versus 91.82% word accuracy using word-level modelling. The results thus lend credence to modelling words by phonemes in vision-based gesture recognition.

1.2 Contributions

The main contributions of the present research are as follows. First, our approach models gestures in terms of their phonemic elements to yield an algorithm that recognizes gesture movement primitives given data captured with a single video camera. Second, our approach uses the apparent motion of an unmarked hand as input as opposed to fitting a model of a hand (arm) or using a mechanical device (e.g. data glove, magnetic tracker). Third, our recognition scheme is based on a nearest neighbour match to prototype signatures, where each of 14 movement primitives of ASL is found to have a distinctive prototype signature in a kinematic feature space. We have evaluated our approach empirically with 592 video sequences and find an 86.00% phoneme accuracy rate for fully automated processing and 97.13% for manually initialized processing even as other aspects of the gesture (hand shape and location) vary.

1.3 Outline of Paper

This paper is subdivided into four main sections. This first section has provided motivation for modelling gestures at the phoneme level. Section 2 describes the linguistic-basis of our representation as well as the algorithmic aspects of the approach. Section 3 documents empirical evaluation of our algorithm instantiation. Finally, Section 4 provides a summary.

2 Technical Approach

Our approach to gesture recognition centres around two main ideas. First, linguistic theory can be used to define a representational substrate that systematically decomposes complex gestures into primitive components. Second, it is desirable to recover the primitives from data that is acquired with a standard video camera and minimal constraints on the user. Currently, we are focused on the recovery of the linguistically defined rigid single handed movement primitives of American Sign Language (ASL). The input is a temporal sequence of images that depicts a single movement phoneme. The output of our system is a classification of the depicted gesture as arising from one of the primitive movements, irrespective of other considerations (e.g., irrespective of hand location and shape). The location of the hand in the initial frame is obtained through an automated localization process utilizing the conjunction of temporal change and skin colour. We assume that the hand is the dominant moving object in the imaged scene as an aid to localization. To affect the recognition, a robust, affine motion estimator is applied to regions of interest defined by skin colour and temporal change on a frame-to-frame basis. The resulting time series of affine parameters are individually accumulated across the sequence to yield a signature that is used for classification of the depicted gesture. Details of the movement gesture vocabulary and the processing stages are presented next.

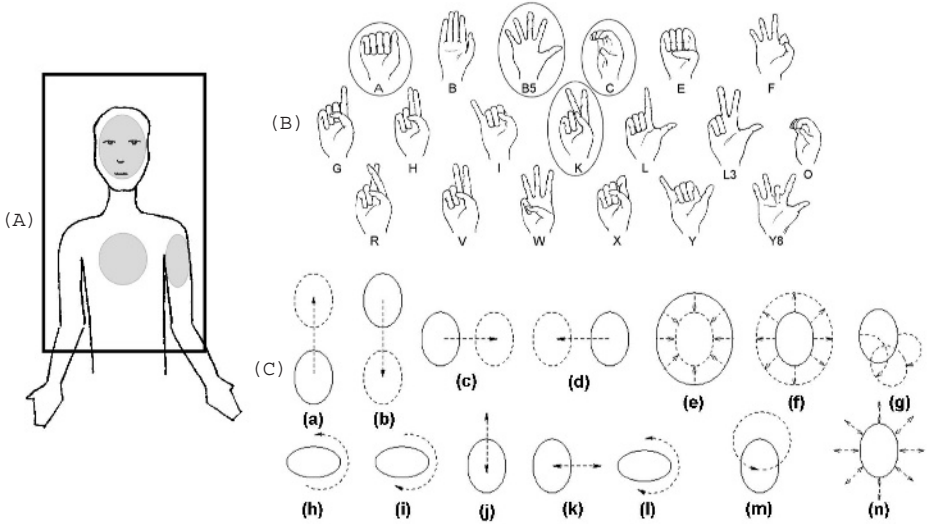


Fig. 1. Stokoe’s phonemic analysis of ASL. The left panel (A) depicts the signing space where the locations reside. Shaded regions indicate locations used in our experiments. The upper right panel (B) depicts possible hand shapes. Circled shapes indicate shapes used in our experiments. The lower right panel (C) depicts possible single handed movements (a) upward (b) downward (c) rightward (d) leftward (e) toward signer (f) away signer (g) nod (h) supinate (i) pronate (j) up and down (k) side to side (l) twist wrist (m) circular (n) to and fro. The solid ellipse, dashed ellipse and dashed arrow represent the initial hand location, the final location and the path taken respectively. We investigate the recognition of movement independent of location and shape.

2.1 Linguistics Basis

Prior to William Stokoe’s seminal work in ASL [27], it was assumed by linguists that the sign was the basic unit of ASL. Stokoe redefined the basic unit of a sign to units analogous to speech phonemes: minimally contrastive patterns that distinguish the symbolic vocabulary of a language. Stokoe’s system consists of three parameters that are executed simultaneously to define a gesture, see Fig. 1. The three parameters capture location, hand shape and movement. There are 12 elemental locations defined by Stokoe residing in a volume in front of the signer termed the “signing space”. The signing space is defined as extending from just above the head to the hip area in the vertical axis and extending close to the extents of the signer’s body in the horizontal axis (see Fig. 1A). There are 19 possible hand shapes (see Fig. 1B). While Stokoe’s complete vocabulary of movements consists of 24 primitives (i.e. single and two-handed movements), as a starting point, we restrict consideration to the 14 rigid *single handed* movements, shown in Fig. 1C. Current ASL theories still recognize the Stokoe system’s basic parameters but differ in their definition of the constituent elements of the parameters [29]. We use Stokoe’s definition of the parameters since they are gen-

erally agreed to represent an important approximation to the somewhat wider and finer grained space that might be required to capture all the subtleties of hand gesture languages.

2.2 Motion Estimation

Let $I(\mathbf{x}, t)$ represent the image brightness at position $\mathbf{x} = (x, y)^\top$ and time t . Using the brightness constancy constraint [12], we define the inter-frame motion, $\mathbf{u}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}))^\top$, as,

$$I(\mathbf{x}, t + 1) = I(\mathbf{x} - \mathbf{u}(\mathbf{x}), t) \quad (1)$$

We employ an affine model to describe the motion,

$$u(x, y) = a_0 + a_1x + a_2y, \quad v(x, y) = a_3 + a_4x + a_5y \quad (2)$$

We make use of the affine model for two main reasons. First, through an analytic derivation we found that there exists a unique mapping between Stokoe's qualitative description of the movement of the hand in the world and the first-order kinematic decomposition of the corresponding visual motion fields. The first-order kinematic description includes the following measures, (differential) translation, rotation, isotropic expansion/contraction and shear: Cases (shown in Fig. 1C) a-d, j, k and m are characterized by translation, for m horizontal and vertical translation oscillate out of phase (see Fig. 2); cases h, i and l involve rotation; cases e, f and n are characterized by expansion/contraction; case g involves shear and contraction. Due to space considerations the derivation has been omitted, for details see [8]. Second, over the small angular extent that encompasses the hand at comfortable signing distances from a camera, small movements can be approximated with an affine model. To affect the recovery of the affine parameters we make use of a robust, hierarchical, gradient-based motion estimator [4] operating over a Gaussian pyramid [15]. The hierarchical nature of the estimator allows us to handle significant magnitude image displacements with computational efficiency even while avoiding local minima. This estimator is applied to skin colour defined regions of interest in a pair of images under consideration. We use skin colour to restrict consideration to image data that arises from the hand; such regions are extracted using a Bayesian maximum-likelihood classifier [32]. As a further level of robustness we restrict consideration to points that experience a significant change in intensity (i.e. dI/dt). For robustness in motion estimation, we make use of an M-estimator [13] (e.g., as opposed to a more standard least-squares approach, c.f., [3]) to allow for operation in the presence of outlying data in the form of non-hand pixels due to skin-colour oversegmentation, pixels that grossly violate the affine approximation as well as points that violate brightness constancy. The particular error norm we choose is the Geman-McClure [13].

The motion estimator is applied to adjacent frames across an image sequence. As an initial seed, the hand region in the first frame of the sequence is outlined by an automated process that consists of: utilizing the conjunction of skin colour

detection and change detection (i.e. dI/dt) to define a map of likely regions where the hand may reside, followed by a morphologically-based shape analysis [15] for the hand itself that seeks the region within the skin/change map containing the maximum circular area. No manual intervention is present. Upon recovering the motion between the first pair of frames, the analysis window is moved based on the affine parameters found (initialized identically to zero at the first frame), the affine parameters are used as the initial parameters for the motion estimation of the next pair of images and the motion estimation process is repeated. When the motion estimator reaches the end of the image sequence, six time series, each representing an affine parameter over the length of the sequence, are realized.

2.3 Kinematic Features

Owing to their descriptive power in the current context, it is advantageous to rewrite the affine parameters in terms of kinematic quantities corresponding to horizontal and vertical translation, divergence, curl and deformation (see, e.g., [16]). In particular, from the coefficients in the affine transformation (2) we calculate the following time series,

$$\begin{aligned} hor(t) &= a_0(t) \\ ver(t) &= a_3(t) \\ div(t) &= a_1(t) + a_5(t) \\ curl(t) &= -a_2(t) + a_4(t) \\ def(t) &= \sqrt{(a_1(t) - a_5(t))^2 + (a_2(t) + a_4(t))^2} \end{aligned} \quad (3)$$

Each of the kinematic time series (3) has an associated unit of measurement (e.g. horizontal/vertical motion are in pixel units) that may differ amongst each other. To facilitate comparisons across the time series for the purposes of recognition, a rescaling of responses is appropriate. We make use of min-max rescaling [11], defined as,

$$\hat{z} = \left(\frac{z - min_1}{max_1 - min_1} \right) \times (max_2 - min_2) + min_2 \quad (4)$$

with min_1 and max_1 the minimum and maximum values (resp.) in the input data z , while min_2 and max_2 specifying the range of the rescaled data taken over the entire population sample. For scaling ranges, we select $[-1, 1]$ for elements of (3) that range symmetrically about the origin and $[0, 1]$ for those with one sided responses, i.e., def .

To complete the definition of our kinematic feature set, we accumulate parameter values across each of the five rescaled kinematic time series, $\hat{hor}(t)$, $\hat{ver}(t)$, $\hat{div}(t)$, $\hat{curl}(t)$, $\hat{def}(t)$ and express each resulting value as a proportion. The accumulation procedure is motivated by the observation that there are two fundamentally different kinds of movements in the vocabulary defined in Fig. 1: those that entail constant sign movements, i.e., movements (a-i), which are uni-directional; those that entail periodic motions, i.e., movements (j-n), which move

“back and forth”. To distinguish these differences, we accumulate our parameter values in two fashions.

First, to distinguish constant sign movements, we compute a *summed response*, SR_i ,

$$SR_i = \sum_{t=1}^T p_{i,t}$$

where $i \in \{\hat{h}or, \hat{v}er, \hat{d}iv, \hat{c}url, \hat{d}ef\}$ indexes a time series, T represents the number of frames a gesture spans and $p_{i,t}$ represents the value of (rescaled) time series i at time t . Constant sign movements should yield non-zero magnitude SR_i , for some i ; whereas, periodic movements will not as their changing sign responses will tend to cancel across time.

Second, to distinguish periodic movements, we compute a *summed absolute response*, SAR_i ,

$$SAR_i = \sum_{t=1}^T |\overline{p_{i,t}}|; \text{ where } \overline{p_{i,t}} = p_{i,t} - mean_i$$

where $mean_i$ represents the mean value of (rescaled) time series i . Now, constant sign movements will have relatively small SAR_i , for all i (given removal of the mean, assuming a relatively constant velocity); whereas, periodic movements will have significantly non-zero responses as the subtracted mean should be near zero (assuming approximate symmetry in the underlying periodic pattern) and the absolute responses now sum to a positive quantity.

Due to the min-max rescaling (4), the SR_i and SAR_i calculated for any given gesture sequence are expressed in comparable ranges on an absolute scale established from consideration of all available data (i.e., min_1 and max_1 are set based on scanning across the entire sample set). For the evaluation of any given gesture sequence, we need to represent the amount of each kinematic quantity observed relative to the others in that particular sequence. For example, a (e.g., very slow) vertical motion in the absence of any other motion should be taken as significant irrespective of the speed. To capture this notion, we convert the accumulated SR_i and SAR_i values to proportions by dividing each computed value by the sum of its consort, formally,

$$SRP_i = SR_i / (\sum_k |SR_k|), \quad SARP_i = SAR_i / (\sum_k SAR_k) \quad (5)$$

with k ranging over $\hat{h}or, \hat{v}er, \hat{d}iv, \hat{c}url, \hat{d}ef$. Here, SRP_i represents the *summed response proportion* of SR parameter i and $SARP_i$ represents the *summed absolute response proportion* of SAR parameter i . Notice that the min-max rescaling accomplished through (4) and the conversion to proportions via (5) accomplish different goals, both of which are necessary: the former brings all the kinematic variables into generally comparable units; the latter adapts the quantities to a given gesture sequence. In the end, we have a 10 component feature set SRP_i and $SARP_i$, $i \in \{\hat{h}or, \hat{v}er, \hat{d}iv, \hat{c}url, \hat{d}ef\}$ that encapsulates the kinematics of the imaged gesture.

Table 1. Gesture signatures. Each movement phoneme has a distinctive prototype signature defined in terms of our kinematic feature set. Kinematic features and movement phonemes are plotted along vertical and horizontal axes, resp. The SRP and SARP values are defined with respect to formula (5).

	SRP									SARP				
	<i>upward</i>	<i>downward</i>	<i>rightward</i>	<i>leftward</i>	<i>toward signer</i>	<i>away signer</i>	<i>supinate</i>	<i>pronate</i>	<i>nod</i>	<i>up and down</i>	<i>side to side</i>	<i>to and fro</i>	<i>twist wrist</i>	<i>circular</i>
<i>hor</i>	0	0	+1	-1	0	0	0	0	0	0	1	0	0	.5
<i>ver</i>	-1	+1	0	0	0	0	0	0	0	1	0	0	0	.5
<i>div</i>	0	0	0	0	+1	-1	0	0	-.5	0	0	1	0	0
<i>curl</i>	0	0	0	0	0	0	+1	-1	0	0	0	0	1	0
<i>def</i>	0	0	0	0	0	0	0	0	+.5	0	0	0	0	0

2.4 Prototype Gesture Signatures

Given our kinematic feature set, each of the primitive movements for ASL, shown in Fig. 1C has a distinctive idealized signature based on (separate) consideration of the SRP_i and $SARP_i$ values (see Table 1). Analytical relationships between the 2D kinematic signatures and the 3D hand movements are presented in [8].

Distinctive signatures for the constant sign movements (i.e., movements a-i in Fig. 1C) are defined with reference to the SRP_i values. Upward/downward movements result in responses to $ver(t)$ alone; hence, of all the SR_i , only $SR_{v\hat{e}r}$ should have a nonzero value in (5), leading to a signature of $|SRP_{v\hat{e}r}| = 1$ while $|SRP_i| = 0, i \neq v\hat{e}r$. In order to disambiguate between upward and downward movements, the sign of $SRP_{v\hat{e}r}$ is taken into account, positive sign for downward and negative for upward. Similarly, rightward/leftward movements result in significant response to $hor(t)$ alone, with the resulting signature of $|SRP_{h\hat{o}r}| = 1$ while $|SRP_i| = 0, i \neq h\hat{o}r$ and positive and negative signed $SRP_{h\hat{o}r}$ corresponding to rightward and leftward movements, resp. The toward/away signer movements are manifest as significant responses in $div(t)$ alone. Correspondingly, $|SRP_{d\hat{i}v}| = 1$ while other values are zero. For this case, positive sign on $SRP_{d\hat{i}v}$ is indicative of toward, while negative sign indicates away. The supinate/pronate gestures map to significant responses in $curl(t)$ alone. Here, $|SRP_{c\hat{u}rl}| = 1$ while other values are zero with positively and negatively signed $SRP_{c\hat{u}rl}$ indicating supinate and pronate, resp. Unlike the other movements described so far, nod has two significant kinematic quantities which have constant signed responses throughout the gesture, namely $def(t)$ and $div(t)$. The sign of $def(t)$ should be positive, while the sign of $div(t)$ should be negative, i.e., contraction. Further, the magnitudes of these two nonzero quantities should be equal. Therefore, we have $|SRP_{d\hat{i}v}| = |SRP_{d\hat{e}f}| = 0.5$ with all other responses zero.

For periodic movements (i.e., movements j-n in Fig. 1C) distinctive signatures are defined with reference to the $SARP_i$ values. The definitions unfold

analogously to those for the constant sign movements, albeit sign now plays no role as the $SARP_i$ are all positive by construction. An up and down movement maps directly to $ver(t)$, resulting in a value of $SARP_{ver}$ equal to 1 with other summed absolute response proportions zero. The side to side movement directly maps to $hor(t)$, resulting in a value of $SARP_{hor}$ equal to 1 while other values are zero. The to and fro movement maps directly to $div(t)$, resulting in a value of $SARP_{div}$ equal to 1 with other summed absolute response proportions zero. The twist wrist movement directly maps to $curl(t)$, resulting in a value of $SARP_{curl}$ equal to 1 with other values zero. The circular movement has two prominent kinematic quantities, $hor(t)$ and $ver(t)$. As the hand traces a circular trajectory, these two quantities will oscillate out of phase with each other (see Fig. 2). Across a complete gesture the two summed absolute responses are equal. The overall signature is thus $SARP_{hor} = SARP_{ver} = 0.5$, with all other values zero.

For classification, we first calculate the Euclidean distance between our input signatures (i.e. SRP_i and $SARP_i$) and their respective stored prototypical signatures. The result is a set of distances d_j (14 in total). Taking the smallest distance as the classified gestures is not sufficient, since it presupposes that we know whether the classification is to be done with respect to the SRP_i (constant sign cases) or the $SARP_i$ (periodic cases). This ambiguity can be resolved through re-weighting the distances by the reciprocal norm of their respective feature vectors, formally,

$$\begin{aligned}\tilde{d}_j &= (1/|\mathbf{SR}|) \times d_j; \text{ where } j \in \{\text{constant sign distance}\} \\ \tilde{d}_j &= (1/|\mathbf{SAR}|) \times d_j; \text{ where } j \in \{\text{periodic distances}\}\end{aligned}$$

with

$$\begin{aligned}\mathbf{SR} &= (SR_{hor}, SR_{ver}, SR_{div}, SR_{curl}, SR_{def}) \\ \mathbf{SAR} &= (SAR_{hor}, SAR_{ver}, SAR_{div}, SAR_{curl}, SAR_{def})\end{aligned}$$

Intuitively, if the norm of \mathbf{SR} is greater than that of \mathbf{SAR} , then the movement is more likely to be a constant sign; if the relative magnitudes are reversed then the movement is more likely to be a periodic. Following the re-weighting, the movement with the smallest \tilde{d}_j value is returned as the classification. Finally, for movements classified by distance as nod, we explicitly check to make sure $|SRP_{div}| \approx |SRP_{def}|$, if not we take the next closest movement. Similarly, for circular we enforce that $SARP_{hor} \approx SARP_{ver}$. These explicit checks serve to reject misclassifications when noise happens to artificially push estimated feature value patterns toward the nod and circular signatures.

3 Empirical Evaluation

To test the viability of our approach, we have tested a software realization of our algorithm on a set of video sequences each of which depicts a human volunteer executing a single movement phoneme. Here, our goal was to test the ability of our algorithm to correctly recognize movement, irrespective of the volunteer, hand location and shape of the complete gesture. Owing to the descriptive power of the phonemic decomposition of gestures into movement, location and shape

primitives, consideration of all possible combinations would lead to an experiment that is not feasible.¹ Instead, we have chosen to subsample the hand shape and location dimensions by exploiting similarities in their respective configurations. For location we have selected whole head, torso and upper arm, see Fig. 1A. These choices allow a range of locations to be considered and also introduce interesting constraints on how movements are executed. For instance, when the hand begins at the upper arm location, the natural tendency is to have the wrist rotated such that the hand is at a slight angle away from the body; as the hand moves towards the opposite side of the body, a slight rotation is introduced to bring the hand roughly parallel with the camera. For hand shape, we have selected A, B5, K and C, see Fig. 1B. The rationale for selecting hand shapes A, B5 and K is as follows: A (i.e. fist) and B5 (i.e. open flat hand) represent the two extremes of the hand shape space, whereas K (i.e. victory sign) represents an approximate midpoint of the space. Hand shape C has been included since it is a clear example of a hand shape being non-planar. This sampling leaves us with a total possible number of test cases equal to 14 (movements) \times 3 (locations) \times 4 (shapes) = 168. However, several of these possibilities are difficult to realize (e.g., pronating movement at the upper arm location); so, dropping these leaves us with a total of 148 cases. Three volunteers each executed all 148 movements while their actions were recorded with a video camera to yield an experimental test set of $3 \times 148 = 444$. In addition, 12 volunteers executed an approximate equal subset of the gesture space (approximately 14 gestures each). In total our experimental test set consisted of 592 gestures. It should be noted that the volunteers were fully aware of the camera and their expected position with respect to it, this allowed precise control of the experimental variables for a systematic empirical test. With an eye toward applications such control is not unrealistic: A natural signing conversation consists of directing one's signing towards the other signer (in this case a camera). During acquisition, standard indoor, overhead fluorescent lighting, was used and the normal (somewhat cluttered) background in our lab was present as volunteers signed in the foreground. Each gesture sequence was captured at a resolution of 640×480 pixels at 30 frames per second; for processing, the gesture sequences were subsampled temporally by a factor of two resulting in a frame rate of 15 frames per second. Typically, the hand region encompasses a region in a frame with dimensions approximately 100 pixels in both width and height. On average the gesture sequences spanned 40 frames for constant sign movements and 80 for periodic movements. Prior to conducting the gesture each volunteer was verbally described the gesture. This was done in order to ensure the capture of naturally occurring extraneous motions which can appear when an unbiased person performs the movements. See Fig. 2 for an example sequence.

¹ Using Stokoe's parameter definitions there would be 14 (movements) \times 19 (shapes) \times 12 (locations) = 3192 combinations for each volunteer.

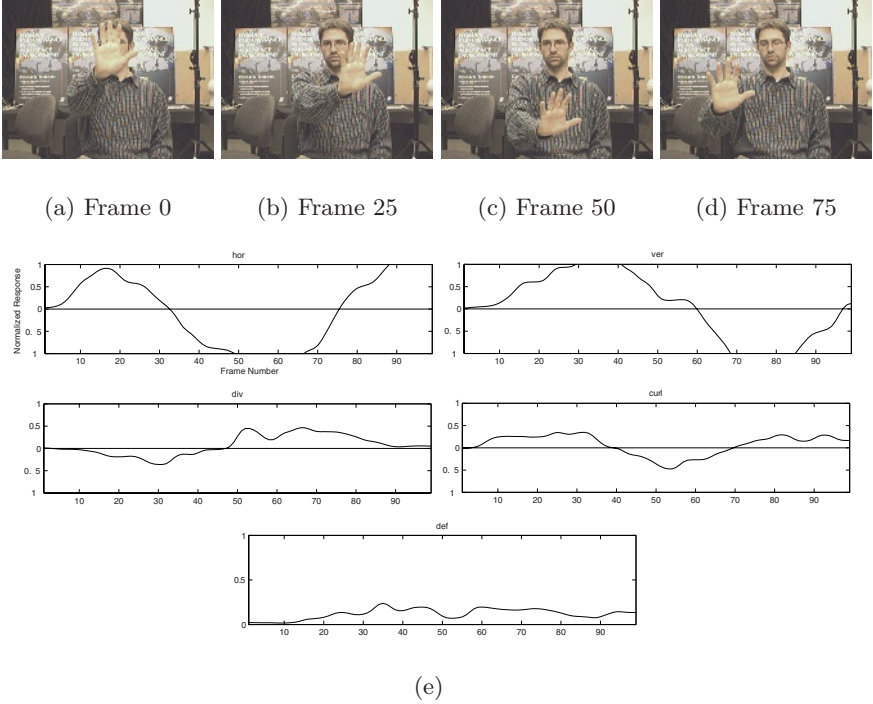


Fig. 2. Circular movement example. A circular movement image sequence with its accompanying kinematic time series plotted. The frame numbers marked on the graphs correspond to the frame numbers of the image sequence.

3.1 Results

To assess the joint performance of the tracker and classification stages, we conducted two trials. The first trial consisted of the hand region being manually outlined in the initial frame and the second trial consisted of the automated initial localization scheme outlined in this paper. In the manually segmented trials 97.13% of the 592 test cases were correctly identified, when considering the top two candidate movements classification performance improved to 99.49%. While for the automated localization trial an accuracy rate of 86.00% was achieved and 91.00% when considering the top two candidates. Further inspection of the results found that approximately 14% of the test cases in the automated localization trial failed to isolate a sufficient region of the hand (i.e. approximately 50% of the hand). The majority of these cases consisted of the automated localization process homing in on the volunteer's head since the head was the dominant moving structure. This is contrary to our assumption that the hand is the dominant moving structure in the scene. Treating these cases as failure to acquire and omitting them from further analysis resulted in an accuracy rate of 91.55% and an accuracy of 95.09% when considering the top two candidates,

Table 2. Gesture movement recognition results. The axes of the table represent the actual input gesture (vertical) versus the classification result (horizontal). Each cell (i,j) in the table holds the percentage of test cases that were actually i but classified as j for both manually initialized localized trials (left) and automated initialized localized trials (right) (i.e. manual/automated). The diagonal (i,j) (highlighted in bold) represents the percentage of the correctly classified gestures.

	<i>up</i>	<i>down</i>	<i>up and down</i>	<i>rightward</i>	<i>leftward</i>	<i>side to side</i>	<i>toward signer</i>	<i>away signer</i>	<i>to and fro</i>	<i>supinate</i>	<i>pronate</i>	<i>twist wrist</i>	<i>nod</i>	<i>circular</i>
<i>up</i>	100 / 92	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/3	0/0	0/0	0/5	0/0	0/0
<i>down</i>	0/0	100 / 91	0/0	0/0	0/0	0/0	0/0	0/0	0/7	0/2	0/0	0/0	0/0	0/0
<i>up and down</i>	0/0	0/0	100 / 95	0/0	0/0	0/3	0/0	0/0	0/0	0/0	0/0	0/3	0/0	0/0
<i>rightward</i>	0/0	0/0	0/0	100 / 92	0/0	0/4	0/0	0/0	0/4	0/0	0/0	0/0	0/0	0/0
<i>leftward</i>	0/0	0/0	0/0	0/0	97 / 85	0/0	0/0	0/0	0/10	0/0	0/0	0/0	3/3	0/3
<i>side to side</i>	0/0	0/0	0/0	0/0	0/0	100 / 86	0/0	0/0	0/0	0/0	0/0	0/11	0/3	0/0
<i>toward signer</i>	0/0	0/0	0/0	0/0	0/0	0/0	96 / 93	0/0	0/0	0/3	0/0	0/3	4/0	0/0
<i>away signer</i>	0/0	2/0	0/0	0/0	0/0	0/0	0/0	98 / 97	0/3	0/0	0/0	0/0	0/0	0/0
<i>to and fro</i>	0/0	0/3	0/0	0/0	0/0	0/0	0/3	0/0	92 / 84	0/0	0/0	0/6	0/3	8/0
<i>supinate</i>	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	97 / 95	0/0	0/3	3/3	0/0
<i>pronate</i>	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	100 / 98	0/0	0/2	0/0
<i>twist wrist</i>	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/2	0/2	0/0	100 / 90	0/5	0/0
<i>nod</i>	0/0	6/0	0/0	0/0	0/0	0/0	3/0	0/0	6/3	0/0	0/0	0/3	84 / 93	0/0
<i>circular</i>	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/2	0/0	0/0	0/7	100 / 91

see Table 2. In terms of execution speed, the tracking speed using a Pentium 4 2.1 GHz processor and unoptimized C code was 8 frames/second; the time consumed by all other components was negligible.

3.2 Discussion

A current limitation is the automated initial localization process. The majority of the failed localization cases were attributed to gross head movements, the remaining localization problems occurred with users gesturing with bare arms (although most bare arm cases were localized properly) and users wearing skin toned clothing. A review of the literature finds that most other related work has simplified the initial localization problem through manual segmentation [19,25, 30], restricting the colours in the scene [17,24,26], restricting the type of clothing worn (i.e. long sleeved shirts) [17,24,26], having users hold markers [5], using a priori knowledge of initial gesture pose [9,14], and using multiple, specially configured cameras [30] or magnetic trackers [6,10,18,30]. In our study, we make no assumptions along these lines; nevertheless, our results are competitive with those reported elsewhere. Beyond initialization, four failed tracking cases occurred related to frame-to-frame displacement beyond the capture range of our motion estimator. Drift has not been a significant factor in tracking during our experiments. This is due to the use of skin colour and change detection masks to define the region of support as well as a robust motion estimator to reject outliers. Possible solutions to tracking failure include: the use of a higher frame

rate camera to decrease interframe motion and/or the use of a motion estimator with a larger capture range (e.g., correlation-based, rather than gradient-based method).

Given acceptable tracking, problems in the classification per se arose from non-intentional but significant movements accompanying the intended movement. For instance, when conducting the “away signer” movement, some of the subjects, would rotate the palm of their hand about the camera axis as they were moving their hand forward. Systematic analysis of such cases may make it possible to improve our feature signatures to encompass such variations.

It should be noted that to realize the above results we assumed that the gestures were temporally segmented. To relax these assumptions future work may appeal to detecting discontinuities in the kinematic feature time series to temporally segment the gestures (e.g. [23]).

4 Summary

We have presented a novel approach to vision-based gesture recognition, based on two key concepts. First, we appeal to linguistic theory to represent complex gestures in terms of their primitive components. By working with a finite set of primitives, which can be combined in a wide variety of ways, our approach has the potential to deal with a large vocabulary of gestures. Second, we define distinctive signatures for the primitive components that can be recovered from monocular image sequences. By working with signatures that can be recovered without special purpose equipment, our approach has the potential for use in a wide range of human computer interfaces. Using American Sign Language (ASL) as a test bed application, we have developed an algorithm for the recognition of the primitive contrastive movements (movement phonemes) from which ASL symbols are built. The algorithm recovers kinematic features from an input video sequence, based on an affine decomposition of the apparent motion(s) across the sequence. The recovered feature values affect movement signatures that are used in a nearest neighbour recognition system. Empirical evaluation of the algorithm suggests its applicability to the analysis of complex gesture videos.

Acknowledgements. The authors thank Antonia Vezos for the illustrations in Fig. 1. Research was funded by the Institute of Robotics and Intelligent Systems one of the government of Canada’s Networks of Centres of Excellence. K. G. Derpanis holds a National Sciences and Engineering Research Council of Canada PGS B fellowship. J. K. Tsotsos holds the Canada Research Chair in Computational Vision.

References

1. J.K. Aggarwal and Q. Cai. Human motion analysis: A review. *CVIU*, 73(3):428–440, 1999.
2. N. Badler. Temporal scene analysis: Conceptual descriptions of object movements. In *Dept. of Comp. Sc., Univ. of Toronto, Rep. TR-80*, 1975.
3. J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV*, pages I:5–10, 1992.
4. M.J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *ICCV*, pages 231–236, 1993.
5. M.J. Black and A.D. Jepson. A probabilistic framework for matching temporal trajectories. In *ECCV*, pages II:909–924, 1998.
6. A.F. Bobick and A.D. Wilson. A state-based approach to the representation and recognition of gesture. *PAMI*, 19(12):1325–1337, Dec 1997.
7. T. Darrell and A. Pentland. Space-time gestures. In *CVPR*, pages 335–340, 1993.
8. K.G. Derpanis. Vision based gesture recognition within a linguistics framework. Master's thesis, York University, Toronto, Canada, 2003.
9. A. Elgammal, V. Shet, Y. Yacoob, and L.S. Davis. Learning dynamics for exemplar-based gesture recognition. In *CVPR*, pages I: 571–578, 2003.
10. S.S. Fels and G.E. Hinton. Glove-talk II. *Trans. on NN*, 9(1):205–212, 1997.
11. J. Han and M. Kamber. *Data Mining*. Morgan Kaufmann, San Francisco, CA, 2001.
12. B.K.P. Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.
13. P.J. Huber. *Robust Statistical Procedures*. SIAM Press, Philadelphia, PA, 1977.
14. M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.
15. B. Jahne. *Digital Image Processing*. Springer, Berlin, 1991.
16. J.J. Koenderink and A.J. van Doorn. Local structure of movement parallax of the plane. *JOSA-A*, 66(7):717–723, 1976.
17. H.K. Lee and J.H. Kim. An HMM-based threshold model approach for gesture recognition. *PAMI*, 21(10):961–973, Oct 1999.
18. R.H. Liang and M. Ouhyoung. A real-time continuous gesture recognition system for sign language. In *AFGR*, pages 558–567, 1998.
19. S. Lu, D. Metaxas, D. Samaras, and J. Oliensis. Using multiple cues for hand tracking and model refinement. In *CVPR*, pages II: 443–450, 2003.
20. V.I. Pavlovic, R. Sharma, and T.S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *PAMI*, 19(7):677–695, July 1997.
21. H. Poizner, U. Bellugi, and V. Lutes-Driscoll. Perception of American Sign Language in dynamic point-light displays. *J. of Exp. Psych.*, 7(2):430–440, 1981.
22. L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.
23. Y. Rui and P. Anandan. Segmenting visual actions based on spatio-temporal motion patterns. In *CVPR*, pages I: 111–118, 2000.
24. J. Schlenzig, E. Hunter, and R. Jain. Vision based gesture interpretation using recursive estimation. In *Asilomar Conf. on Signals, Systems and Computers*, 1994.
25. C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3d human tracking. In *CVPR*, pages I: 69–76, 2003.
26. T. Starner, J. Weaver, and A.P. Pentland. Real-time American Sign Language recognition using desk and wearablecomputer based video. *PAMI*, 20(12):1371–1375, December 1998.

27. W.C. Stokoe, D. Casterline, and C. Croneberg. *A Dictionary of American Sign Language*. Linstok Press, Washington, DC, 1965.
28. J.K. Tsotsos, J. Mylopoulos, H.D. Covvey, and S.W. Zucker. A framework for visual motion understanding. *PAMI*, 2(6):563–573, November 1980.
29. C. Valli and C. Lucas. *Linguistics of American Sign Language: An Introduction*. Gallaudet University Press, Washington, D.C., 2000.
30. C. Vogler and D. Metaxas. A framework for recognizing the simultaneous aspects of American Sign Language. *CVIU*, 81(3):358–384, 2001.
31. M.H. Yang, N. Ahuja, and M. Tabb. Extraction of 2d motion trajectories and its application to hand gesture recognition. *PAMI*, 24(8):1061–1074, August 2002.
32. B. Zarit, B.J. Super, and F. Quek. Comparison of five color models in skin pixel classification. In *RATFG*, pages 58–63, 1999.

Line Geometry for 3D Shape Understanding and Reconstruction

Helmut Pottmann, Michael Hofer, Boris Odehnal, and Johannes Wallner

Technische Universität Wien, A 1040 Wien, Austria.

{pottmann,hofer,odehnal,wallner}@geometrie.tuwien.ac.at

Abstract. We understand and reconstruct special surfaces from 3D data with line geometry methods. Based on estimated surface normals we use approximation techniques in line space to recognize and reconstruct rotational, helical, developable and other surfaces, which are characterized by the configuration of locally intersecting surface normals. For the computational solution we use a modified version of the Klein model of line space. Obvious applications of these methods lie in Reverse Engineering. We have tested our algorithms on real world data obtained from objects as antique pottery, gear wheels, and a surface of the ankle joint.

Introduction. The geometric viewpoint turned out to be highly successful in dealing with a variety of problems in Computer Vision (see, e.g., [3,6,9,15]). So far mainly methods of analytic geometry (projective, affine and Euclidean) and differential geometry have been used. The present paper suggests to employ *line geometry* as a tool which is both interesting and applicable to a number of problems in Computer Vision. Relations between vision and line geometry are not entirely new. Recent research on generalized cameras involves sets of projection rays which are more general than just bundles [1,7,18,22]. A beautiful exposition of the close connections of this research area with line geometry has recently been given by T. Pajdla [17].

The present paper deals with the problem of *understanding and reconstructing 3D shapes from 3D data*. The data are assumed to be of a surface-like nature — either a cloud of measurement points, or another 3D shape representation such as a triangular mesh or a surface representation in parametric or implicit form — and we assume that we are able to obtain a discrete number of points on the surface and to estimate surface normals there. We are interested in classes of surfaces with special properties: planar, spherical and cylindrical surfaces, surfaces of revolution and helical surfaces; and the more general surface classes of *canal*, *pipe*, and *developable* surfaces. For applications in CAD/CAM it is essential that such special shapes are not represented by freeform surfaces without regard to their special properties, but treated in a way more appropriate to their ‘simple’ nature.

Line geometry enters the problem of object reconstruction from point clouds via the surface normals estimated at the data points. In fact, modern 3D photography and the corresponding software delivers such normals together with the data points. It turns out that the surface classes mentioned above can be

characterized in terms of their surface normals in an elegant, simple and computationally efficient way. Appropriate coordinates for lines (which yield a *model of line space* as a certain 4-dimensional manifold embedded in \mathbb{R}^6) will allow to classify point clouds and their normals (i.e., the so-called ‘normal congruence’) by means of tools such as principal component analysis.

Previous work. There is a vast body of literature on surface reconstruction. Since we are interested in the reconstruction of special surfaces, we do not review the part of literature which deals with the reconstruction of triangular meshes or general freeform representations. In Computer Vision, recognition and reconstruction of *special shapes* is often performed by methods related to the *Hough transform*. Originally designed for the detection of straight lines in 2D, it received much attention and has been generalized so as to be able to detect and reconstruct many other shapes (see, e.g., [11,14]). Pure Hough transform methods work in ‘spaces of shapes’ and quickly lead to high dimensions and reduced efficiency. In order to avoid these problems, such tools are sometimes augmented by methods from constructive geometry. This approach is already close to techniques invented by the CAD community, which use geometric characterizations of surfaces (e.g. by means of the Gaussian image) for data segmentation and the extraction of special shapes (see the survey [24]). Many papers deal with axis estimation of rotational surfaces, like [25]. See [8] for an overview on the Hough transform, the RANSAC principle, and the least squares approach. In the present paper, however, rotational surfaces occur only as a special case.

The use of line geometry for surface reconstruction has been introduced by [20]. There cylinders, surfaces of revolution and helical surfaces are recognized by the fact that their surface normals are contained in a so-called linear line complex. In particular surfaces which can be moved within themselves in more than one way (right circular cylinders, spheres, planes) are detected. The technique is extendable to surfaces which may be locally well approximated by the surface types mentioned above [2,13,21].

Contributions of the present paper. Inspired by the line geometric work on reverse engineering of special shapes, our paper presents a broader line geometric framework for the solution of problems in 3D shape understanding, segmentation and reconstruction:

- We discuss a point model for line space, namely a certain 4-dimensional algebraic manifold M^4 of order 4 in \mathbb{R}^6 . It is better suited for line geometric approximation problems than the classical Klein model or the model used in [20], which is limited to linear line complexes.
- This point model makes it possible to perform the basic shape recognition tasks via principal component analysis (PCA) of a point cloud (contained in M^4), which represents the estimated surface normals of the input shape. This procedure is further improved here and unlike [20] is stable in all special cases.
- The idea of looking for surface normals which *intersect* makes it possible to apply line-geometric methods to the problem of recognition and reconstruction

of *canal surfaces* (which are the envelope of a one-parameter family of spheres), and of *moulding surfaces*. The latter denotes a certain class of sweep surfaces which contains the developable surfaces and the pipe surfaces.

- The segmentation of composite surfaces into their ‘simple’ parts is addressed here in so far as recognition of surface type is essential for segmentation. Our algorithms may be included as a ‘black box’ into a segmentation algorithm.

1 The 4-Dimensional Manifold of Lines in \mathbb{R}^6

This paragraph discusses a computationally attractive point model of (i.e., coordinates in) the 4-dimensional manifold of straight lines in space. It is closely related to the classical Klein quadric (i.e., Plücker coordinates). We think of an *oriented* line L as one equipped with a unit vector \mathbf{l} indicating the direction of the line — so that there are two oriented lines for each line of space. Then L is determined by \mathbf{l} and the moment vector $\bar{\mathbf{l}}$, which is computed by means of an arbitrary point \mathbf{x} on L as $\bar{\mathbf{l}} = \mathbf{x} \times \mathbf{l}$. $\bar{\mathbf{l}}$ is independent of the choice of \mathbf{x} on L . The six numbers $(\mathbf{l}, \bar{\mathbf{l}})$ are called *normalized Plücker coordinates* of L . ‘Normalized’ means that $\|\mathbf{l}\| = 1$; further, they satisfy the orthogonality condition $\mathbf{l} \cdot \bar{\mathbf{l}} = 0$. Conversely, any two vectors $\mathbf{l}, \bar{\mathbf{l}} \in \mathbb{R}^3$ which satisfy these two conditions determine a unique oriented straight line L in \mathbb{R}^3 , which has $(\mathbf{l}, \bar{\mathbf{l}})$ as its normalized Plücker coordinates.

If we do not distinguish between the two opposite orientations of the same line, we may use all multiples of the pair $(\mathbf{l}, \bar{\mathbf{l}})$ as coordinates of a line. Of course, we still have the condition $\mathbf{l} \cdot \bar{\mathbf{l}} = 0$. Such homogeneous coordinate vectors of lines represent those points of five-dimensional projective space which are contained in the *Klein quadric* M_2^4 given by the equation $(\mathbf{x}, \bar{\mathbf{x}}) \in M_2^4 \Leftrightarrow \mathbf{x} \cdot \bar{\mathbf{x}} = 0$. This interpretation of lines is well studied in classical geometry, see [21].

The present paper pursues the following approach, which is closely related to the Klein quadric. We use only normalized coordinate vectors, and so we identify an oriented line with the point $(\mathbf{l}, \bar{\mathbf{l}})$ in six-dimensional Euclidean space \mathbb{R}^6 . In this way we obtain a mapping α of oriented lines to points of a 4-dimensional manifold $M^4 \subset \mathbb{R}^6$. M^4 is algebraic of degree 4, and is the intersection of the cylinder Z^5 and the cone Γ^5 defined by

$$Z^5 : \mathbf{x}^2 = 1, \quad \Gamma^5 : \mathbf{x} \cdot \bar{\mathbf{x}} = 0.$$

We use the Euclidean distance of points in \mathbb{R}^6 in order to measure distances between oriented lines G, H : If $G\alpha = (\mathbf{g}, \bar{\mathbf{g}})$ and $H\alpha = (\mathbf{h}, \bar{\mathbf{h}})$, then

$$d(G, H)^2 = (\mathbf{g} - \mathbf{h})^2 + (\bar{\mathbf{g}} - \bar{\mathbf{h}})^2. \quad (1)$$

The reasons why we prefer this distance function are the following: On the one hand, it is quadratic and thus lends itself to minimization. On the other hand, in a neighbourhood of the origin of the coordinate system, (1) models a distance of lines which is in accordance with visualization. The slight drawback that this is no longer the case in regions far away from the origin is in fact not important in most applications, as such applications very often define a natural *region of interest*, where we can put the origin of our coordinate system.

Remark 1. Another method for introducing coordinates and measuring distances for lines, which has been used in the past, is to fix two parallel planes and describe a line by the two intersection points with that plane (cf. [21]). This leads to simpler formulae and a region of interest which is bounded by the two fixed planes (in our case, this region is a sphere centered in the origin).

Classification of Surfaces by Normal Congruences

The set of normals of a surface is called its *normal congruence*. Some surface types are easily recognized from properties of their normal congruence: The normals of a sphere pass through its center (they constitute a bundle with a finite vertex), and the normals of a plane are parallel (they constitute a parallel bundle, with vertex at infinity). These are the simplest examples; there are however other interesting and practically important classes of surfaces which are nicely characterized by their normal congruence. These classical results are basic to shape understanding and reconstruction algorithms and thus we summarize them in this section.

A uniform motion in 3-space, composed of a uniform rotation of unit angular velocity about an axis and a translation of constant speed p along this axis is called a *helical motion of pitch p* . If we choose a Cartesian coordinate system with the x_3 -axis being the axis of rotation, then the point (x_1, x_2, x_3) will move according to

$$x_1(t) = x_1 \cos t - x_2 \sin t, x_2(t) = x_1 \sin t + x_2 \cos t, \quad x_3(t) = x_3 + pt. \quad (2)$$

In the case $p = 0$ we have a uniform *rotation*. For $p \rightarrow \infty$ we get, in the limit, a uniform *translation*. A surface swept by a curve under a helical motion is called *helical surface*. As special and limit cases for $p = 0$ and $p = \infty$ we get *surfaces of revolution* and *cylinder surfaces*, respectively. We say that these surfaces are *kinematically generated*. However if we speak of a helical surface we always mean *part* of a complete helical surface as defined above, and analogously for other adjectives, like rotational/cylindrical/spherical/planar. Closely related to helical motions are *linear complexes*, which are certain three-parameter sets of lines defined by linear equations, and which are discussed in more detail in [21]. A line L with Plücker coordinates $(\mathbf{l}, \bar{\mathbf{l}})$ is contained in the complex \mathcal{C} with coordinates $(\mathbf{c}, \bar{\mathbf{c}})$ if and only if

$$L \in \mathcal{C} \iff \bar{\mathbf{c}} \cdot \mathbf{l} + \mathbf{c} \cdot \bar{\mathbf{l}} = 0. \quad (3)$$

Obviously the lines of the complex \mathcal{C} defined by (3) correspond to those points of \mathbb{R}^6 which are both contained in M^4 and fulfill (3), i.e., they lie in M^4 and in the hyperplane $H^5 : \bar{\mathbf{c}} \cdot \mathbf{x} + \mathbf{c} \cdot \bar{\mathbf{x}} = 0$. Note that H^5 passes through the origin. The set $\mathcal{C}\alpha$ is a certain 3-dimensional manifold. A linear complex \mathcal{C} is called *singular* if $\mathbf{c} \cdot \bar{\mathbf{c}} = 0$ (then there is a line A with $A\alpha = (\mathbf{c}, \bar{\mathbf{c}})$, and \mathcal{C} consists of all lines which intersect A).

The sets of lines which correspond to the intersections of M^4 with other d -dimensional subspaces H^d also play important roles and have special names.¹ These line sets are employed by the following classification result (see e.g. [21])

Proposition 1. *The normals of a given surface are contained in a linear complex $\mathcal{C} = (\mathbf{c}, \bar{\mathbf{c}})$ if and only if the surface is helical or rotational or cylindrical. The complex \mathcal{C} is regular for a helical surface and singular otherwise. The axis of \mathcal{C} is at infinity (i.e., $\mathbf{c} = 0$) for cylindrical surfaces. Parts of surfaces which are kinematically generated in more than one way are characterized as follows: The normals of a right circular cylinder are contained in a linear congruence; the normals of spherical and planar surfaces are contained in a bundle.*

A list of surface classes and their normals. Surfaces of revolution (Fig. 1f) are envelopes of a one-parameter family of spheres, the centers of which lie on the axis of rotation. More generally the envelope of a smooth one-parameter family of spheres is called a *canal surface* (see Fig. 1a). The midpoints of these spheres form the surface's *spine curve*. If a sphere touches the envelope surface it does so along a circle.

If the spheres are of constant radius, one obtains a *pipe surface* (see Fig. 1c). Pipe and canal surfaces appear in CAD for example as blending surfaces. Obviously pipe surfaces play an important role in places like oil platforms and refineries. As-built reconstructions of these shapes from 3D data have a number of applications and thus this topic already received some attention (see e.g. [16]). Viewpoint-invariant recognition of canal surfaces from images has been addressed e.g. by Pillow et al. [19].

The envelope of a sphere under rotation about an axis (not through the sphere's center) is a *torus* (Fig. 1g), which may be generated as a canal surface also in another way. In general, surfaces, which are canal surfaces in two ways, are called *Dupin cyclides* (see Fig. 1e). They are well known algebraic surfaces of degree 3 or 4, and may be described as images of tori under inversion. Their use in geometric modelling is described e.g. in a survey article by W. Degen [4].

Pipe surfaces are also traced out by a circle \mathbf{p} which moves such that its center runs along the spine curve $\mathbf{s}(t)$, and its plane $U(t)$ is always orthogonal to that curve. Rotation about the spine curve has no influence on the resulting surface, but we would like to specify the movement in a more precise way: We assume that a coordinate frame attached to the circle moves such that its angular velocity is minimal (i.e., no unnecessary rotations about the spine curve occur; see e.g. [12]). This is the case of the *rotation-minimizing frame*, where trajectories

¹ For $d = 4$ we get *linear line congruences* (e.g. a *hyperbolic* linear congruence consists of all lines which meet two given lines). The case $d = 3$ consists of three subcases: (i) the *bundle* of lines incident with a given point; (ii) the *field* of lines contained in a given plane; and (iii) a *regulus*, which is one of the two one-parameter families of lines contained in a ruled quadric. In the cases (i) and (ii) H^3 is contained in Γ^5 ; whereas in case (iii) the set $\Gamma^5 \cap H^3$ is a quadratic cone. For $d = 2$ it may happen that H^2 is contained in Γ^5 , in which case the corresponding line set is a *pencil*, consisting of lines in a fixed plane U which pass through a point $p \in U$.

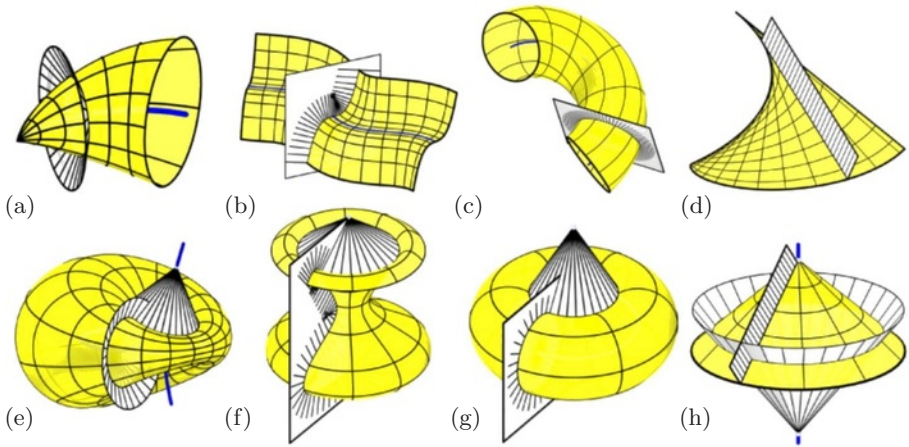


Fig. 1. (a) Canal surface, (b) Moulding surface, (c) Pipe surface, (d) Developable surface, (e) Dupin cyclide, (f) Surface of revolution, (g) Torus, (h) Cone of revolution. Both the surface and the set $I(\mathbf{x})$ are shown.

of points in the plane $U(t)$ are orthogonal to $U(t)$. A surface generated by any profile curve \mathbf{p} (not necessarily a circle) during this motion is called a *moulding surface* (see Fig. 1b). It has the property that all positions $\mathbf{p}(t)$ of the profile curve are principal curvature lines.

If the profile curve \mathbf{p} is a straight line, the moulding surface is *developable* and can be mapped isometrically into the Euclidean plane. All points of a *generator* or *ruling* $\mathbf{p}(t)$ have the same tangent plane. Thus, such a surface is also the envelope of a one-parameter family of planes. Special cases are cones and cylinders, but in general the lines $\mathbf{p}(t)$ are tangent to some space curve (the *line of regression*), which is a singular curve on the surface (see Fig. 1d).

Let us now describe how to characterize these surfaces via their normal congruences. It is well known that the normal congruence of any surface Φ , which is not planar or spherical, can be decomposed into two families of developable surfaces. These are formed by the surface normals along the principal curvature lines. For canal surfaces one of these families consists of cones (or possibly cylinders) of revolution, and the vertices of these cones lie on the spine curve (see Fig. 1a). For pipe surfaces, these cones have an opening angle of 180 degrees (i.e., they are line pencils; see Fig. 1c). For developable surfaces the cones become cylinders of infinite radius (i.e., they are pencils of parallel lines; see Fig. 1d).

In order to exploit these facts computationally (cf. Sec. 3), we define the set $I(\mathbf{x})$ of *locally intersecting normals* at a point \mathbf{x} of a surface Φ : We pick a neighborhood $N(\mathbf{x}) \subset \Phi$ of \mathbf{x} and look for points \mathbf{y} in $N(\mathbf{x})$ whose normals intersect the normal at \mathbf{x} . These normals form the set $I(\mathbf{x})$. Interesting are those cases where we can identify two specially shaped components of $I(\mathbf{x})$.

Proposition 2. *The following table enumerates special surface classes and the shapes of the two components of $I(\mathbf{x})$ for any surface point \mathbf{x} (cf. Fig. 1).*

	<i>cone</i>	<i>planar</i>	<i>pencil</i>	<i>parallel</i>
<i>arbitrary</i>	<i>canal surface</i>	<i>moulding surface</i>	<i>pipe surface</i>	<i>developable surface</i>
<i>cone</i>	<i>cyclide</i>	<i>rotational surface</i>	<i>torus</i>	<i>rotational cone</i>

Conversely, if for all \mathbf{x} we can identify two components of $I(\mathbf{x})$ as listed, the surface is of the associated type.

Here, ‘arbitrary’ means that no condition is imposed on this component, ‘parallel’ means that the lines of the component are parallel, and ‘planar’ means that the lines are contained in some plane. The table can be used as follows. If, for example, we find that $I(\mathbf{x})$ contains, for all \mathbf{x} , a parallel component, and a conical component, the surface is a cone of revolution (as a side-effect the conical component itself must be part of a cone of revolution; see Fig. 1h). Note that the second row of the table, except for the cyclides, contains only rotational surfaces already addressed by Prop. 1. This is the reason why adding new rows in the table above does not lead to anything new.

2 Principal Component Analysis (PCA) on the Surface Normals for Surface Recognition and Reconstruction

Basic algorithm and PCA. We would like to approximate a set of lines L_i , $i = 1, \dots, N$ by a linear line complex \mathcal{C}^* with coordinates $(\mathbf{c}^*, \bar{\mathbf{c}}^*)$. It will be sufficient to consider one orientation for each line, so we compute normalized Plücker coordinates $L_i\alpha = (\mathbf{l}_i, \bar{\mathbf{l}}_i) \in \mathbb{R}^6$.

A first approximation method is to compute the unique hyperplane H^5 through the origin which minimizes the squared sum of distances from the points $L_i\alpha$. With $H(\mathbf{x}, \bar{\mathbf{x}}) = \bar{\mathbf{c}} \cdot \mathbf{x} + \mathbf{c} \cdot \bar{\mathbf{x}}$, the Euclidean distance of a point $(\mathbf{p}, \bar{\mathbf{p}})$ from the hyperplane $H^5 : H(\mathbf{x}, \bar{\mathbf{x}}) = 0$ simply equals $H(\mathbf{p}, \bar{\mathbf{p}})$, if

$$\mathbf{c}^2 + \bar{\mathbf{c}}^2 = 1. \tag{4}$$

Thus we find H^5 by minimizing $F(\mathbf{c}, \bar{\mathbf{c}}) = \sum_{i=1}^N [\bar{\mathbf{c}} \cdot \mathbf{l}_i + \mathbf{c} \cdot \bar{\mathbf{l}}_i]^2$ under the constraint (4). We consider both \mathbf{c} and $\bar{\mathbf{c}}$ as column vectors and write F in the form

$$F(\mathbf{c}, \bar{\mathbf{c}}) = [\mathbf{c}^T \ \bar{\mathbf{c}}^T] \cdot A \cdot \begin{bmatrix} \mathbf{c} \\ \bar{\mathbf{c}} \end{bmatrix},$$
 with a certain symmetric 6×6 -matrix A . It follows

easily from the Lagrangian multiplier rule that the minimizer $(\mathbf{c}^*, \bar{\mathbf{c}}^*)$ is given by an eigenvector of A , which belongs to A ’s smallest eigenvalue λ^* , and which is normalized according to (4) (in this case we also have $F(\mathbf{c}^*, \bar{\mathbf{c}}^*) = \lambda^*$).

According to Prop. 1, to a linear complex \mathcal{C} belongs a helical motion. Its pitch p and the Plücker coordinates $(\mathbf{a}, \bar{\mathbf{a}})$ of its axis can be computed by $p = (\mathbf{c}^* \cdot \bar{\mathbf{c}}^*)/(\mathbf{c}^*)^2$, $(\mathbf{a}, \bar{\mathbf{a}}) = (\mathbf{c}^*, \bar{\mathbf{c}}^* - p\mathbf{c})$ (cf. [21]). A small pitch p indicates that the lines L_i belong to a surface of revolution. We may then repeat the approximation process with the additional side condition that $\mathbf{c} \cdot \bar{\mathbf{c}} = 0$ (according to Prop. 1). Large values of p indicate that the lines L_i belong to a cylinder. This circumstance is also detected by the fact that all L_i are orthogonal to a fixed line. If we decide we want the data to be approximated by a cylinder, we may repeat the minimization with the additional side condition that $\mathbf{c} = 0$.

So far we have shown how to find the *motion* which generates the surface. For the actual computation of a *profile curve* we use methods available in the literature [20,21,23,24].

Two small eigenvalues of A indicate that the points $L_i\alpha \in \mathbb{R}^6$ are almost contained in a subspace of dimension 4. The normals L_i are then almost contained in a linear congruence. By Prop. 1 this means that the normals belong to a *right circular cylinder*. As a side-effect, all of them must now intersect the cylinder's axis orthogonally.

Three small eigenvalues of A imply that the points $L_i\alpha$ are almost contained in a 3-dimensional subspace, which indicates a spherical or planar surface. Note however that planes are also detected by a Gaussian image [24] or a Hough transform for planes in 3-space.

Remark 2. The procedure above is a principal component analysis of the set of image points $L_i\alpha \in \mathbb{R}^6$. Without going into details we remark that it may be refined in order to be more robust e.g. by incorporating a weighted least squares approximation which iteratively downweights outliers, or the RANSAC principle (see [5] and references therein).

Refined algorithm. So far we computed an approximating complex \mathcal{C} by finding a hyperplane H^5 (carrying $\mathcal{C}\alpha$) such that the sum of squares of distances of the points $L_i\alpha$ from H^5 is minimized. This is not the same as minimizing distances of the points $L_i\alpha$ from the set $\mathcal{C}\alpha = M^4 \cap H^5$. Actually we should have minimized the *geodesic distance* of the points $L_i\alpha$ from the set $\mathcal{C}\alpha = H^5 \cap M^4$ within the manifold M^4 . We will take this into account by considering the angle of intersection between H^5 and M^4 . For a point $(\mathbf{l}, \bar{\mathbf{l}}) = L\alpha$ in M^4 consider M^4 's tangent 4-space T^4 and let $\phi = \angle(T^4, H)$. d_H denotes the distance of $L\alpha$ from H^5 . Then the value $d_{\mathcal{C}} = d_H / \sin \phi$ is an estimate for the geodesic distance of $L\alpha$ from $\mathcal{C}\alpha = M^4 \cap H^5$. There is the following result:

Lemma 1. *The angle ϕ used in the definition of $d_{\mathcal{C}}$ is given by $\cos^2 \phi = (\bar{\mathbf{c}} \cdot \mathbf{l})^2 + (\bar{\mathbf{c}} \cdot \bar{\mathbf{l}}^n + \mathbf{c} \cdot \mathbf{l}^n)^2$, with $(\mathbf{l}^n, \bar{\mathbf{l}}^n) = 1/(\mathbf{l}^2 + \bar{\mathbf{l}}^2) \cdot (\mathbf{l}, \bar{\mathbf{l}})$.*

Proof. We note that the 2-dimensional normal space to $M^4 = Z^5 \cap \Gamma^5$ in the point $L\alpha = (\mathbf{l}, \bar{\mathbf{l}})$ is spanned by the normal vectors \mathbf{a} to Z^5 and \mathbf{b} to Γ^5 . These are given by $\mathbf{a} = (\mathbf{l}, \mathbf{o})$ and $\mathbf{b} = (\bar{\mathbf{l}}, \mathbf{l})$, respectively. Note that $\mathbf{a} \cdot \mathbf{b} = 0$, and that $\|\mathbf{a}\| = 1$, whereas \mathbf{b} is not yet a unit vector in \mathbb{R}^6 . In order to achieve this, we define \mathbf{l}^n and $\bar{\mathbf{l}}^n$ as above, and replace \mathbf{b} by $\mathbf{b}^n = (\bar{\mathbf{l}}^n, \mathbf{l}^n)$. Now $(\mathbf{c}, \bar{\mathbf{c}})$ is a normal vector of H^5 and satisfies (4). It follows that $\cos^2 \phi = (\mathbf{a} \cdot \mathbf{n})^2 + (\mathbf{b}^n \cdot \mathbf{n})^2$, and the result follows by expanding the definitions. \square

Minimizing the distances $d_{\mathcal{C}}$ instead of orthogonal distances d_H means minimizing the function

$$F_1(\mathbf{c}, \bar{\mathbf{c}}) = \sum_{i=1}^N \frac{(\bar{\mathbf{c}} \cdot \mathbf{l}_i + \mathbf{c} \cdot \bar{\mathbf{l}}_i)^2}{1 - (\bar{\mathbf{c}} \cdot \mathbf{l}_i)^2 - (\bar{\mathbf{c}} \cdot \bar{\mathbf{l}}_i^n + \mathbf{c} \cdot \mathbf{l}_i^n)^2}$$

under the side-condition (4). This is a nonlinear optimization problem which we solve via a weight iteration: We initialize the algorithm with the minimizer

of F . Then, using the solution $(\mathbf{c}^*, \bar{\mathbf{c}}^*)$ from the previous step, we minimize $F_2(\mathbf{c}, \bar{\mathbf{c}}) = \sum_{i=1}^N w_i (\bar{\mathbf{c}} \cdot \mathbf{l}_i + \mathbf{c} \cdot \bar{\mathbf{l}}_i)^2$, with $1/w_i = 1 - (\bar{\mathbf{c}}^* \cdot \mathbf{l}_i)^2 - (\bar{\mathbf{c}}^* \cdot \bar{\mathbf{l}}_i^n + \mathbf{c}^* \cdot \mathbf{l}_i^n)^2$. This procedure is motivated by the fact that different angles ϕ_i at points L_i give different weights to distances $d_{H,i}$ in the minimization of F . The unbalanced situation is corrected via the weight iteration described above. The iteration terminates if the change in F_1 in one iteration step is below some given threshold.

3 Examples

We used a Minolta VI-900 3D laser scanner to obtain point data and tested the effectiveness of our algorithms on them. Point clouds were thinned if necessary and triangulated. Surface normal vectors have been estimated by local regression planes. In Figures 2, 3, and 4, data points have been rendered as small balls.

Surface type by PCA: Rotational surfaces. Fig. 2 shows the procedure of reconstructing the axis and meridian curve of a near-rotational surface. Data have been obtained by scanning the outer surface of a late Hallstatt pottery object manufactured in approx. 550 B.C. without the use of a pottery wheel. The procedure of Sec. 2 has been applied to estimate a linear line complex which fits the surface normals best. With d as the diameter of the point cloud, we found $|p| \ll d$, so the data points in question may be approximated by a surface of revolution. The meridian curve (Fig. 2, center) was found by rotating the point cloud into a plane through the axis of rotation (a) and approximating (a) by a smooth curve (b). The deviations of the original cloud from the reconstructed ideal surface of revolution are shown by Fig. 2, right. Figure 2, left, shows how the estimated axis (3) changes if we use only parts of the data available (dotted lines 1,2). As is to be expected, accuracy decreases if we use small meridian strips.

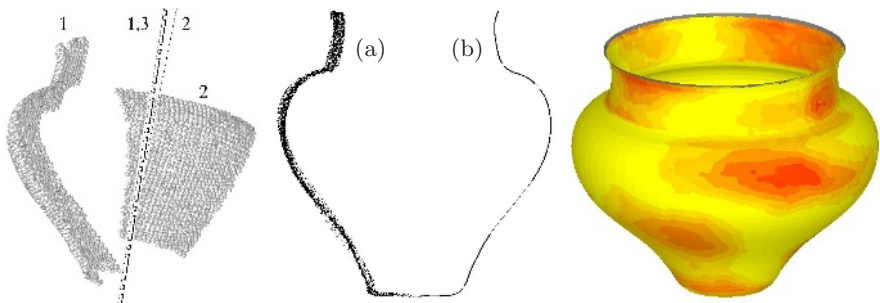


Fig. 2. Reconstruction of axis and meridian curve of a pottery object. Left: Axes reconstructed from parts (1,2) and from entire object (3). Center: Reconstructed meridian curve. Right: Colour coded deviations of original from reconstructed ideal surface (maximum deviation: 2.6% of diameter).

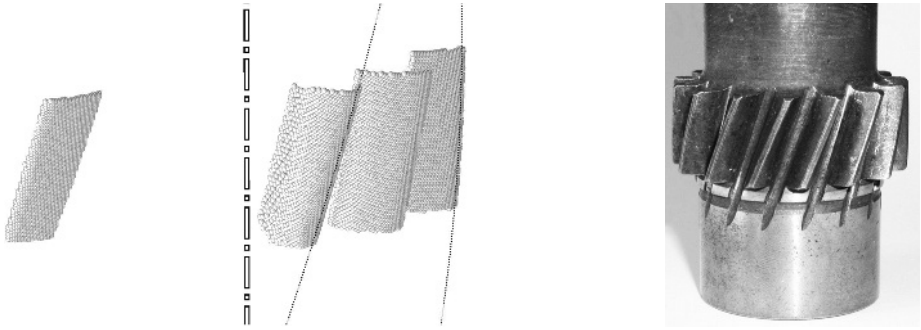


Fig. 3. Left: Reconstruction of axis and two helical paths for helical gears. Right: Photo of helical gears.

Surface type by PCA: Helical surfaces. Fig. 3, left shows a scan of four teeth of a helical gear wheel as an example of a surface which is known to be helical. The point data have been obtained in one pass of scanning. The underlying helical motion, defined by axis and parameter, has been reconstructed in the expected way.

Surface type by PCA: Freeform surfaces. The human body does not possess mathematically exact helical surfaces. However, we studied the following interesting example: Fig. 4, left, shows a scan of a *trochlea tali*, i.e., the distal interface of the ankle joint. The closest helical surface computed by the algorithm of Sec. 2 is not a surface of revolution. This piece of information is important when studying the relative motion of the talus (ankle bone) with respect to the tibiofibular (lower leg) system. We might ask whether the trochlea tali is close enough to a mathematical helical surface to be called helical. This turns out to be not the case, as can be seen from computing the closest helical surfaces to surface strips and comparing the results. The axes corresponding to four strips together with the axis corresponding to the entire data set are indicated in Fig. 4.

Surface type by $I(\mathbf{x})$. In order to recognize even more surface classes, we consider the sets $I(\mathbf{x})$ of locally intersecting normals. We compute them as follows: The surface normal spanned by the point \mathbf{x}' and the vector \mathbf{n}' is contained in $I(\mathbf{x})$, if and only if $\det(\mathbf{x}' - \mathbf{x}, \mathbf{n}, \mathbf{n}') = 0$ (\mathbf{n} being the normal vector at \mathbf{x}). The discrete version of this is that the normals at the endpoints of an edge of a triangulation of our point cloud are in $I(\mathbf{x})$, if the determinant mentioned above changes its sign along that edge.

Our examples — scans of plaster and wood models — are shown by Fig. 4, where the data points whose normals contribute to $I(\mathbf{x})$ are indicated in black. We can see that $I(\mathbf{x})$ consists of two components, one of which may be an arc (in case of a canal surface) or a straight line (in case of a developable surface).

In order to apply Prop. 2, we have to test whether one or more of the two components of $I(\mathbf{x})$ are conical or planar. If $L_i = (\mathbf{l}_i, \bar{\mathbf{l}}_i)$ ($i = 1, \dots, N$) are the

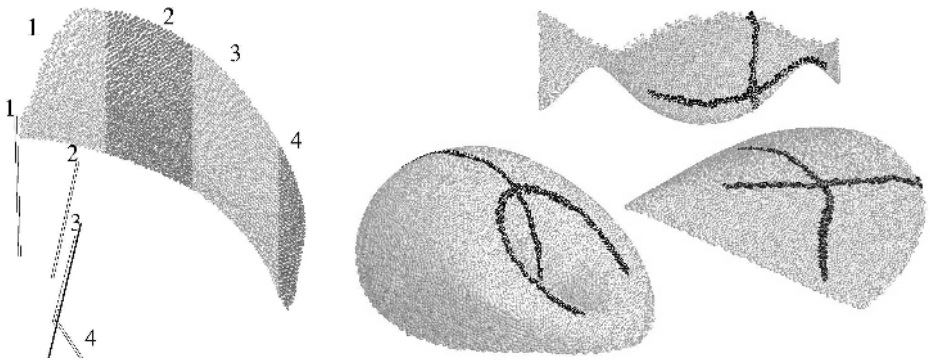


Fig. 4. Left: Computation of axes of nearest helical surfaces for strips 1–4 and also the entire data set (shown in bold) of a trochlea tali. Right: The sets $I(\mathbf{x})$ for, counting clockwise from top, a surface of revolution, the developable *oid*, and for the plaster model of the Dupin cyclide shown in [10], Fig. 229a.

surface normals of such a component, we have to find either a point incident with all of them or a plane which contains them. There are several ways to do this. One would be to consider the lines L_i as surface normals and to try to reconstruct the corresponding surface type according to Sec. 2. This, however, would lead to numerical difficulties due to thinness of data. Therefore we recognized conical components with vertex \mathbf{s} in the following way: Incidence of the point \mathbf{s} with the line $(\mathbf{l}_i, \bar{\mathbf{l}}_i)$ is characterized by $\bar{\mathbf{l}}_i = \mathbf{s} \times \mathbf{l}_i$. Thus finding a point \mathbf{s} ‘as incident as possible’ with the lines L_i means minimizing

$$G(\mathbf{x}) = \sum w_i (\bar{\mathbf{l}}_i - \mathbf{s} \times \mathbf{l}_i)^2, \quad (w_i \geq 0) \quad (5)$$

with weights w_i which determine the influence of the single lines. Minimizing (5) is standard, and iteratively downweighting outliers makes this method stable. Comparison of the minimal value of \sqrt{G} with the size of the point cloud determines conicality. *Planar* components of $I(\mathbf{x})$ are detected in a similar way. Note that line pencils are both planar and conical.

The sets $I(\mathbf{x})$ of Fig. 4 have, counting clockwise from the top, 1, 0, and 2 (non-planar) conical components, whereas the number of planar components equals 1, 1, and 0, respectively. It follows that these surfaces are a rotational surface, a moulding surface, and a Dupin cyclide. Additional information on the parallelity of surface normals along one component of $I(\mathbf{x})$ shows that the right hand surface shown by Fig. 4 is a developable one.

Conclusion and Future Research. We have shown how techniques from classical line geometry can serve for recognizing and reconstructing special surfaces. Other applications, such as ruled surface approximation and the computation of approximating line congruences can benefit from the use of the embedding of line space into \mathbb{R}^6 . Active B-spline curves and surfaces can be used efficiently.

This is a consequence of the fact that finding footpoints on M^4 corresponds to finding the zeros of a fourth order polynomial, and is in fact equivalent to the footpoint problem for planar ellipses.

Future research will also address how to use the bisecting linear complex (cf. [21], p. 166) for checking and improving point correspondences in 3D registration problems, especially when looking for a good initial position.

Acknowledgements. This work was supported by the Austrian Science Fund (FWF) under grants P16002 and P15911; and the innovative project ‘3D technology’ of Vienna University of Technology. We want to express our thanks to Landesarchäologie Salzburg for the archaeological pottery artifact.

References

1. Benosman R., Kang S. B. (editors): *Panoramic Vision: Sensors, Theory and Applications*, Springer Verlag, 2001.
2. Chen H.-Y., Lee I.-K., Leopoldseder S., Pottmann H., Randrup T., Wallner J.: On surface approximation using developable surfaces, *Graphical Models and Image Processing*, 61:110–124, 1999.
3. Cipolla R., Giblin P.: *Visual Motion of Curves and Surfaces*, Cambridge UP, 2000.
4. Degen W.: Cyclides, In *Handbook of Computer Aided Geometric Design*, G. Farin, J. Hoschek and M.-S. Kim, eds., Elsevier, pp. 575–601, 2002.
5. De la Torre, F., Black M. J.: Robust principal component analysis for Computer Vision, *Proc. 8th Int. Conf. on Computer Vision*, pp. 362–369, 2001.
6. Faugeras O.: *Three-dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Cambridge, MA, 1993.
7. Gupta R., Hartley R.: Linear pushbroom cameras, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):963–975, 1997.
8. Halř R.: *Estimation of the axis of rotation of fragments of archaeological pottery*. Proc. 21st Workshop Austrian Assoc. for Pattern Recognition, Hallstatt 1997.
9. Hartley R., Zisserman A.: *Multiple View Geometry in Computer Vision*, Cambridge Univ. Press, Cambridge, UK, 2000.
10. Hilbert D., Cohn-Vossen S., *Anschauliche Geometrie*. Springer, 1932. Reprinted 1996. Translated as: *Geometry and the Imagination*, American Math. Soc. 1999.
11. Illingworth J., Kittler J.: A survey of the Hough transform, *Computer Vision, Graphics and Image Processing*, 44:87–116, 1988.
12. Jüttler B., Wagner M.: Kinematics and animation, In *Handbook of Computer Aided Geometric Design*, G. Farin et al. eds., Elsevier, pp. 723–748, 2002.
13. Kós G., Martin R., Várady T.: Recovery of blend surfaces in reverse engineering, *Computer Aided Geometric Design*, 17:127–160, 2000.
14. Leavers V.: Which Hough transform? *CVGIP:Image Understanding* 58:250–264, 1993.
15. Mundy J., Zissermann A.: *Geometric Invariance in Computer Vision*, MIT Press, ’92.
16. Navab N.: Canonical representation and three view geometry of cylinders, *Intl. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Commission III, Vol. XXXIV, Part 3A, pp. 218–224, 2002.
17. Pajdla T.: Stereo geometry of non-central cameras, PhD thesis, CVUT, 2002.
18. Peleg S., Ben-Ezra M., Pritch Y.: Omnistereo: panoramic stereo imaging, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):279–290, 2001.

19. Pillow N., Utcke S., Zisserman A.: Viewpoint-invariant representation of generalized cylinders using the symmetry set, *Image Vision Comput.*, 13:355–365, 1995.
20. Pottmann H., Randrup T., Rotational and helical surface reconstruction for reverse engineering, *Computing*, 60:307–322, 1998.
21. Pottmann H., Wallner J.: *Computational Line Geometry*, Springer 2001.
22. Seitz S. M.: The space of all stereo images, *Proceedings ICCV 2001*, pp. 26–33.
23. Várady T., Benkő T., G. Kós T.: Reverse engineering regular objects: simple segmentation and surface fitting procedures, *Int. J. Shape Modeling*, 4:127–141, 1998.
24. Várady T., Martin R.: Reverse Engineering, In *Handbook of Computer Aided Geometric Design*, G. Farin et al., eds., Elsevier, pp. 651–681, 2002.
25. Willis A., Orriols X., Cooper D.: Accurately Estimating Sherd 3D Surface Geometry with Application to Pot Reconstruction. CVPR Workshop, 2003.

Extending Interrupted Feature Point Tracking for 3-D Affine Reconstruction

Yasuyuki Sugaya and Kenichi Kanatani

Department of Information Technology, Okayama University, Okayama
700-8530 Japan, {sugaya,kanatani}@suri.it.okayama-u.ac.jp

Abstract. Feature point tracking over a video sequence fails when the points go out of the field of view or behind other objects. In this paper, we extend such interrupted tracking by imposing the constraint that under the affine camera model all feature trajectories should be in an affine space. Our method consists of iterations for optimally extending the trajectories and for optimally estimating the affine space, coupled with an outlier removal process. Using real video images, we demonstrate that our method can restore a sufficient number of trajectories for detailed 3-D reconstruction.

1 Introduction

The factorization method of Tomasi and Kanade [15] can reconstruct the 3-D shape of a scene from feature point trajectories tracked over a video sequence. The computation is very efficient, requiring only linear operations. The solution is sufficiently accurate for many practical purposes and can be used as an initial value for iterations of a more sophisticated reconstruction procedure [3].

However, the feature point tracking fails when the points go out of the field of view or behind other objects. In order to obtain a sufficient number of feature trajectories for detailed 3-D reconstruction, we need to extend such interrupted tracking to the final frame. There have been several such attempts in the past.

Tomasi and Kanade [15] reconstructed the 3-D positions of partly visible feature points from their visible image positions and reprojected them onto the frames in which they are invisible. The camera positions were estimated from other visible feature points.

Saito and Kamijima [12] projectively reconstructed tentative 3-D positions of the missing points by sampling two frames in which they are visible and then reprojected them onto the frames in which they are invisible. The camera positions were computed up to projectivity.

Using the knowledge that all trajectories of feature points should be in a 4-D subspace of the data space, Jacobs [5] randomly sampled four trajectories, constructed a high-dimensional subspace by letting the missing data have free values, and computed its orthogonal complement. He repeated this many times and computed by least squares a 4-D subspace approximately orthogonal to the

resulting orthogonal complements¹. Partial trajectories were extended so that they were compatible with the estimated subspace. A similar method was also used by Kahl and Heyden [6].

Brandt [1] reconstructed tentative 3-D positions of the missing points using a tentative camera model and reprojected them onto all frames. From the visible and reprojected feature points, he estimated the camera model. Iterating these, he optimized both the camera model and the feature positions.

For all these methods, we should note the following:

- We need not reconstruct a tentative 3-D shape. 3-D reconstruction is made possible by some geometric constraints over multiple frames. One can directly map 2-D point positions to other frames if such constraints² are used.
- If a minimum number of frames are sampled for tentative 3-D reconstruction, the accuracy of computation depends on the sampled frames. Rather, one should make full use of all information contained in all frames.
- The observed trajectories are not necessarily correct, but existing methods treat outlier removal and trajectory extension separately.

In this paper, we present a new scheme for extending partial trajectories based on the constraint that under the affine camera model all trajectories should be in a 3-D affine space, which we call the “affine space constraint”. Our method consists of iterations for optimally extending the trajectories and for optimally estimating the affine space.

If the motion were pure rotation, one could do exact maximal likelihood estimation, e.g., by using the method of Shum et al. [13], but it cannot be applied to translational motions. Here, we simplify the optimization procedure by introducing to each partial trajectory a weight that reflects its length. Also, we incorporate outlier removal and trajectory extension into a single process, testing in every step of the optimization if each trajectory, extended or not, is reliable and removing unreliable ones as outliers.

Thus, the contribution of this paper is as follows:

1. We present a succinct mathematical formulation for extending interrupted trajectories based on the affine space constraint without referring to any particular camera model such as orthography. Our constraint is stronger than that used by Jacobs [5]. No reprojection of tentative 3-D reconstruction is necessary.
2. We present a procedure that integrates reliability evaluation of perfect and imperfect trajectories, outlier removal, and optimization of the affine space into a single process.

Sec. 2 summarizes our affine space constraint. Sec. 3 describes our initial outlier removal procedure. Sec. 4 describes how we extend partial trajectories

¹ In actual computation, he interchanged the roles of points and frames: he sampled two frames, i.e., two lists of x coordinates and two lists of y coordinates. The mathematical structure is the same.

² The projective reconstruction of Saito and Kamijima [12] is equivalent to the use of the *trilinear constraint* [3].

and test their reliability. In Sec. 5, we show real video examples and demonstrate that our method can restore a sufficient number of trajectories for detailed 3-D reconstruction. Sec. 6 presents our conclusion.

2 Affine Space Constraint

We first summarize the geometric constraints on which our method is based. The same constraints have already been used in our previous studies [7,8,9,14]. We reiterate them here, because they play a fundamental role in our trajectory extension method.

Suppose we track N feature points over M frames. Let $(x_{\kappa\alpha}, y_{\kappa\alpha})$ be the coordinates of the α th point in the κ th frame. We stack all the coordinates vertically and represent the entire trajectory by the following $2M$ -D *trajectory vector*:

$$\mathbf{p}_\alpha = (x_{1\alpha} \ y_{1\alpha} \ x_{2\alpha} \ y_{2\alpha} \ \cdots \ x_{M\alpha} \ y_{M\alpha})^\top. \quad (1)$$

For convenience, we identify the frame number κ with “time” and refer to the κ th frame as “time κ ”.

We regard the XYZ camera coordinate system as the world frame, relative to which the scene is moving. Consider a 3-D coordinate system fixed to the scene, and let \mathbf{t}_κ and $\{\mathbf{i}_\kappa, \mathbf{j}_\kappa, \mathbf{k}_\kappa\}$ be, respectively, its origin and basis vectors at time κ . If the α th point has coordinates $(a_\alpha, b_\alpha, c_\alpha)$ with respect to this coordinate system, the position with respect to the world frame at time κ is

$$\mathbf{r}_{\kappa\alpha} = \mathbf{t}_\kappa + a_\alpha \mathbf{i}_\kappa + b_\alpha \mathbf{j}_\kappa + c_\alpha \mathbf{k}_\kappa. \quad (2)$$

We assume an affine camera, which generalizes orthographic, weak perspective, and paraperspective projections [10]: the 3-D point $\mathbf{r}_{\kappa\alpha}$ is projected onto the image position

$$\begin{pmatrix} x_{\kappa\alpha} \\ y_{\kappa\alpha} \end{pmatrix} = \mathbf{A}_\kappa \mathbf{r}_{\kappa\alpha} + \mathbf{b}_\kappa, \quad (3)$$

where \mathbf{A}_κ and \mathbf{b}_κ are, respectively, a 2×3 matrix and a 2-D vector determined by the position and orientation of the camera and its internal parameters at time κ . Substituting Eq. (2), we have

$$\begin{pmatrix} x_{\kappa\alpha} \\ y_{\kappa\alpha} \end{pmatrix} = \tilde{\mathbf{m}}_{0\kappa} + a_\alpha \tilde{\mathbf{m}}_{1\kappa} + b_\alpha \tilde{\mathbf{m}}_{2\kappa} + c_\alpha \tilde{\mathbf{m}}_{3\kappa}, \quad (4)$$

where $\tilde{\mathbf{m}}_{0\kappa}$, $\tilde{\mathbf{m}}_{1\kappa}$, $\tilde{\mathbf{m}}_{2\kappa}$, and $\tilde{\mathbf{m}}_{3\kappa}$ are 2-D vectors determined by the position and orientation of the camera and its internal parameters at time κ . From Eq. (4), the trajectory vector \mathbf{p}_α in Eq. (1) can be written in the form

$$\mathbf{p}_\alpha = \mathbf{m}_0 + a_\alpha \mathbf{m}_1 + b_\alpha \mathbf{m}_2 + c_\alpha \mathbf{m}_3, \quad (5)$$

where \mathbf{m}_0 , \mathbf{m}_1 , \mathbf{m}_2 , and \mathbf{m}_3 are the $2M$ -D vectors obtained by stacking $\tilde{\mathbf{m}}_{0\kappa}$, $\tilde{\mathbf{m}}_{1\kappa}$, $\tilde{\mathbf{m}}_{2\kappa}$, and $\tilde{\mathbf{m}}_{3\kappa}$ vertically over the M frames, respectively.

Eq. (5) implies that all the trajectories are constrained to be in the 4-D subspace spanned by $\{\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$ in \mathcal{R}^{2M} . This is called the *subspace constraint* [7,8], on which the method of Jacobs [5] is based.

In addition, the coefficient of \mathbf{m}_0 in Eq. (5) is identically 1 for all α . This means that the trajectories are in the 3-D affine space within that 4-D subspace. This is called the *affine space constraint* [9].

If all the feature points are tracked to the final frame, we can define the coordinate origin at the centroid of their trajectory vectors³ $\{\mathbf{p}_\alpha\}$, thereby regarding them as defining a 3-D subspace in \mathcal{R}^{2M} . The Tomasi-Kanade factorization [15] is based on this representation, and Brandt [1] tried to find this representation by iterations. In this paper, we directly use the affine space constraint without searching for the centroid.

Unlike existing studies, we describe our trajectory extension scheme without referring to any particular camera model, such as orthographic, weak perspective, or paraperspective projection, except that it is affine. Of course, existing methods described with respect to a particular camera model can automatically be generalized to all affine cameras, but our formulation makes this fact more explicit.

3 Outlier Removal

Before extending partial trajectories, we must remove incorrectly tracked trajectories, or “outliers”, from among observed complete trajectories.

This problem was studied by Huynh and Heyden [4], who fitted a 4-D subspace to the observed trajectories by LMedS [11], removing those trajectories sufficiently apart from it. However, their distance measure was introduced merely for mathematical convenience without giving much consideration to the statistical behavior of image noise.

Sugaya and Kanatani [14] fitted a 4-D subspace to the observed trajectories by RANSAC [2,3] and removed outliers using a χ^2 criterion derived from the error behavior of actual video tracking. Here, we modify their method specifically for the affine space constraint. Our method is a direct consequence of the principle given in [14], but we describe it here, because it plays a crucial role for our optimization procedure we introduce later.

3.1 Procedure

Let $n = 2M$, where M is the number of frames, and let $\{\mathbf{p}_\alpha\}$, $\alpha = 1, \dots, N$, be the observed complete trajectory vectors. Our outlier removal procedure is as follows:

³ If the origin of the scene coordinate system is at the centroid of the feature points, we have $\sum_{\alpha=1}^N a_\alpha = \sum_{\alpha=1}^N b_\alpha = \sum_{\alpha=1}^N c_\alpha = 0$, so we can see from Eq. (5) that \mathbf{m}_0 is at the centroid of the trajectory vectors in \mathcal{R}^{2M} . If we let $\mathbf{p}_\alpha = \mathbf{p}_\alpha - \mathbf{m}_0$, we obtain from Eq. (5) $(\mathbf{p}'_1 \cdots \mathbf{p}'_N) = (\mathbf{m}_1 \mathbf{m}_2 \mathbf{m}_3) \begin{pmatrix} a_1 & \cdots & a_N \\ b_1 & \cdots & b_N \\ c_1 & \cdots & c_N \end{pmatrix}$ or “ $\mathbf{W} = \mathbf{MS}$ ” as commonly described in the literature.

1. Randomly choose four vectors $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$, and \mathbf{q}_4 from among $\{\mathbf{p}_\alpha\}$.
2. Compute the $n \times n$ moment matrix

$$\mathbf{M}_3 = \sum_{i=1}^4 (\mathbf{q}_i - \mathbf{q}_C)(\mathbf{q}_i - \mathbf{q}_C)^\top, \quad (6)$$

where \mathbf{q}_C is the centroid of $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4\}$.

3. Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the three eigenvalues of the matrix \mathbf{M}_3 , and $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ the orthonormal system of corresponding eigenvectors.
4. Compute the $n \times n$ projection matrix

$$\mathbf{P}_{n-3} = \mathbf{I} - \sum_{i=1}^3 \mathbf{u}_i \mathbf{u}_i^\top. \quad (7)$$

5. Let S be the number of points \mathbf{p}_α that satisfy

$$\|\mathbf{P}_{n-3}(\mathbf{p}_\alpha - \mathbf{q}_C)\|^2 < (n-3)\sigma^2, \quad (8)$$

where σ is an estimate of the noise standard deviation.

6. Repeat the above procedure a sufficient number of times⁴, and determine the projection matrix \mathbf{P}_{n-3} that maximizes S .
7. Remove those \mathbf{p}_α that satisfy

$$\|\mathbf{P}_{n-3}(\mathbf{p}_\alpha - \mathbf{q}_C)\|^2 \geq \sigma^2 \chi_{n-3;99}^2, \quad (9)$$

where $\chi_{r;a}^2$ is the a th percentile of the χ^2 distribution with r degrees of freedom.

The term $\|\mathbf{P}_{n-3}(\mathbf{p}_\alpha - \mathbf{q}_C)\|^2$, which we call the *residual*, is the squared distance of point \mathbf{p}_α from the fitted 3-D affine space. If the noise in the coordinates of the feature points is an independent Gaussian random variable of mean 0 and standard deviation σ , the residual $\|\mathbf{P}_{n-3}(\mathbf{p}_\alpha - \mathbf{q}_C)\|^2$ divided by σ^2 should be subject to a χ^2 distribution with $n-3$ degrees of freedom. Hence, its expectation is $(n-3)\sigma^2$. The above procedure effectively fits a 3-D affine space that maximizes the number of the trajectories whose residuals are smaller than $(n-3)\sigma^2$. After fitting such an affine space, we remove those trajectories which cannot be regarded as inliers with significance level 1% (Fig. 1). We have confirmed that the value $\sigma = 0.5$ can work well for all image sequences we tested [14].

3.2 Final Affine Space Fitting

After removing outlier trajectories, we optimally fit a 3-D affine space to the resulting inlier trajectories. Let $\{\mathbf{p}_\alpha\}$, $\alpha = 1, \dots, N$, be their trajectory vectors. We first compute their centroid and the $n \times n$ moment matrix

$$\mathbf{p}_C = \frac{1}{N} \sum_{\alpha=1}^N \mathbf{p}_\alpha, \quad \mathbf{M} = \sum_{\alpha=1}^N (\mathbf{p}_\alpha - \mathbf{p}_C)(\mathbf{p}_\alpha - \mathbf{p}_C)^\top. \quad (10)$$

⁴ In our experiment, we stopped if S did not increase for 200 consecutive iterations.

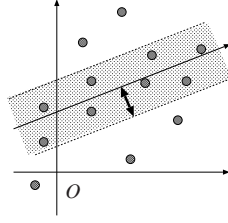


Fig. 1. Removing outliers by fitting a 3-D affine space.

Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the largest three eigenvalues of the matrix \mathbf{M} , and $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ the orthonormal system of corresponding eigenvectors. The optimally fitted 3-D affine space is spanned by the three vectors of \mathbf{u}_1 , \mathbf{u}_2 , and \mathbf{u}_3 starting from \mathbf{p}_C .

Mathematically, this affine space fitting is equivalent to the factorization operation using SVD (singular value decomposition) [15]. It follows that no SVD is necessary for 3-D reconstruction once an affine space is fitted⁵.

4 Trajectory Extension

We now describe our trajectory extension scheme.

4.1 Reliability Test

If the α th feature point can be tracked only over κ of the M frames, its trajectory vector \mathbf{p}_α has $n - k$ unknown components (as before, we put $n = 2M$ and $k = 2\kappa$). We partition the vector \mathbf{p}_α into the k -D part $\mathbf{p}_\alpha^{(0)}$ consisting of the k known components and the $(n - k)$ -D part $\mathbf{p}_\alpha^{(1)}$ consisting of the remaining $n - k$ unknown components. Similarly, we partition⁶ the centroid \mathbf{p}_C and the basis vectors $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ into the k -D parts $\mathbf{p}_C^{(0)}$ and $\{\mathbf{u}_1^{(0)}, \mathbf{u}_2^{(0)}, \mathbf{u}_3^{(0)}\}$ and the $(n - k)$ -D parts $\mathbf{p}_C^{(1)}$ and $\{\mathbf{u}_1^{(1)}, \mathbf{u}_2^{(1)}, \mathbf{u}_3^{(1)}\}$ in accordance with the division of \mathbf{p}_α .

We test if each of the partial trajectories is sufficiently reliable. Let \mathbf{p}_α be a partial trajectory vector. If image noise does not exist, the deviation of \mathbf{p}_α from the centroid \mathbf{p}_C should be expressed as a linear combination of \mathbf{u}_1 , \mathbf{u}_2 , and \mathbf{u}_3 . Hence, there should be constants c_1 , c_2 , and c_3 such that

$$\mathbf{p}_\alpha^{(0)} - \mathbf{p}_C^{(0)} = c_1 \mathbf{u}_1^{(0)} + c_2 \mathbf{u}_2^{(0)} + c_3 \mathbf{u}_3^{(0)} \quad (11)$$

⁵ The statement that the method of Tomasi and Kanade [15] is based on matrix factorization using SVD is not correct. It simply means 3-D affine reconstruction based on the affine camera model. The SVD is merely one of many equivalent computational tools for it.

⁶ This is merely for the convenience of description. In real computation, we treat all data as n -D vectors after multiplying them by an appropriate diagonal matrix consisting of 1s and 0s.

for the known part. In the presence of image noise, this equality does not hold. If we let $\mathbf{U}^{(0)}$ be the $k \times 3$ matrix consisting of $\mathbf{u}_1^{(0)}$, $\mathbf{u}_2^{(0)}$, and $\mathbf{u}_3^{(0)}$ as its columns, Eq. (11) is replaced by

$$\mathbf{p}_\alpha^{(0)} - \mathbf{p}_C^{(0)} \approx \mathbf{U}^{(0)} \mathbf{c}, \quad (12)$$

where \mathbf{c} is the 3-D vector consisting of c_1 , c_2 , and c_3 . Assuming that $k \geq 3$, we estimate the vector \mathbf{c} by least squares in the form

$$\hat{\mathbf{c}} = \mathbf{U}^{(0)-} (\mathbf{p}_\alpha^{(0)} - \mathbf{p}_C^{(0)}), \quad (13)$$

where $\mathbf{U}^{(0)-}$ is the generalized inverse of $\mathbf{U}^{(0)}$. It is computed by

$$\mathbf{U}^{(0)-} = (\mathbf{U}^{(0)\top} \mathbf{U}^{(0)})^{-1} \mathbf{U}^{(0)\top}. \quad (14)$$

The residual, i.e., the squared distance of point $\mathbf{p}_\alpha^{(0)}$ from the 3-D affine space spanned by $\{\mathbf{u}_1^{(0)}, \mathbf{u}_2^{(0)}, \mathbf{u}_3^{(0)}\}$ is $\|\mathbf{p}_\alpha^{(0)} - \mathbf{p}_C^{(0)} - \mathbf{U}^{(0)} \hat{\mathbf{c}}\|^2$. If the noise in the coordinates of the feature points is an independent Gaussian random variable of mean 0 and standard deviation σ , the residual $\|\mathbf{p}_\alpha^{(0)} - \mathbf{p}_C^{(0)} - \mathbf{U}^{(0)} \hat{\mathbf{c}}\|^2$ divided by σ^2 should be subject to a χ^2 distribution with $k - 3$ degrees of freedom. Hence, we regard those trajectories that satisfy

$$\|\mathbf{p}_\alpha^{(0)} - \mathbf{p}_C^{(0)} - \mathbf{U}^{(0)} \hat{\mathbf{c}}\|^2 \geq \sigma^2 \chi_{k-3;99}^2 \quad (15)$$

as outliers with significance level 1%.

4.2 Extension and Optimization of Trajectories

The unknown part $\mathbf{p}_\alpha^{(1)}$ is estimated from the constraint implied by Eq. (11), namely

$$\mathbf{p}_\alpha^{(1)} - \mathbf{p}_C^{(1)} = c_1 \mathbf{u}_1^{(1)} + c_2 \mathbf{u}_2^{(1)} + c_3 \mathbf{u}_3^{(1)} = \mathbf{U}^{(1)} \mathbf{c}, \quad (16)$$

where $\mathbf{U}^{(1)}$ is the $(n - k) \times 3$ matrix consisting of $\mathbf{u}_1^{(1)}$, $\mathbf{u}_2^{(1)}$, and $\mathbf{u}_3^{(1)}$ as its columns. Substituting Eq. (13) for \mathbf{c} , we obtain

$$\hat{\mathbf{p}}_\alpha^{(1)} = \mathbf{p}_C^{(1)} + \mathbf{U}^{(1)} \mathbf{U}^{(0)-} (\mathbf{p}_\alpha^{(0)} - \mathbf{p}_C^{(0)}). \quad (17)$$

Evidently, this is an optimal estimate in the presence of Gaussian noise. However, the underlying affine space is computed only from a small number of complete trajectories; no information contained in the partial trajectories is used, irrespective of how long they are. So, we incorporate partial trajectories by iterations.

Note that if three components of \mathbf{p}_α are specified, one can place it, in general, in any 3-D affine space by appropriately adjusting the remaining $n - 3$ components. In view of this, we introduce the “weight” of the trajectory vector \mathbf{p}_α with k known components in the form

$$W_\alpha = \frac{k - 3}{n - 3}. \quad (18)$$

Let N be the number of all trajectories, complete or partial, inliers or outliers. The optimization goes as follows:

1. Set the weights W_α of those trajectories, complete or partial, that are so far judged to be outliers to 0. All other weights are set to the value in Eq. (18).
2. Fit a 3-D affine space to all the trajectories. The procedure is the same as described in Sec. 3.2 except that Eqs. (10) are replaced by the *weighted* centroid and the *weighted* moment matrix:

$$\mathbf{p}_C = \frac{\sum_{\alpha=1}^N W_\alpha \mathbf{p}_\alpha}{\sum_{\alpha=1}^N W_\alpha}, \quad \mathbf{M} = \sum_{\alpha=1}^N W_\alpha (\mathbf{p}_\alpha - \mathbf{p}_C)(\mathbf{p}_\alpha - \mathbf{p}_C)^\top. \quad (19)$$

3. Test each trajectory if it is an outlier, using Eq. (15).
4. Estimate the unknown parts of the inlier partial trajectory vectors, using Eq. (17).

These four steps are iterated until the fitted affine space converges. Eq. (17) implies that the estimated components do not contribute to the residual of the extended vector \mathbf{p}_α from the affine space, so the reliability of extended trajectories is tested only from their known components using Eq. (15). In the course of this optimization, trajectories once regarded as outliers may be judged to be inliers later, and vice versa. In the end, inlier partial trajectories are optimally extended with respect to the affine space that is optimally fitted to all the complete and partial inlier trajectories.

However, the resulting solution is not guaranteed to be globally optimal; its accuracy largely depends on the quality of the initial guess. The outlier removal procedure of Sec. 3 is incorporated for obtaining as accurate an initial guess as possible, even though all trajectories are reexamined later.

The iterations may not converge if the initial guess is very poor or a large proportion of the trajectories are incorrect. In that case, we must conclude that the original feature tracking does not provide meaningful information. However, this did not happen in any of our experiments using real video sequences.

We need at least three complete trajectories for guessing the initial affine space. If no such trajectories are given, we may use the method of Jacobs [5] for an initial guess. However, it is much more practical to segment the sequence into overlapping blocks, extending partial trajectories over each block separately and connecting all the blocks to find complete trajectories.

5 Experiments

We tested our method using real video sequences. Fig. 2(a) shows five decimated frames from a 50 frame sequence (320×240 pixels) of a static scene taken by a moving camera. We detected 200 feature points and tracked them using the Kanade-Lucas-Tomasi algorithm [16]. When tracking failed at some frame, we restarted the tracking after adding a new feature point in that frame. Fig. 2(b) shows the life space of the 871 trajectories thus obtained: they are enumerated on the horizontal axis in the order of disappearance and new appearance; the white part corresponds to missing data.

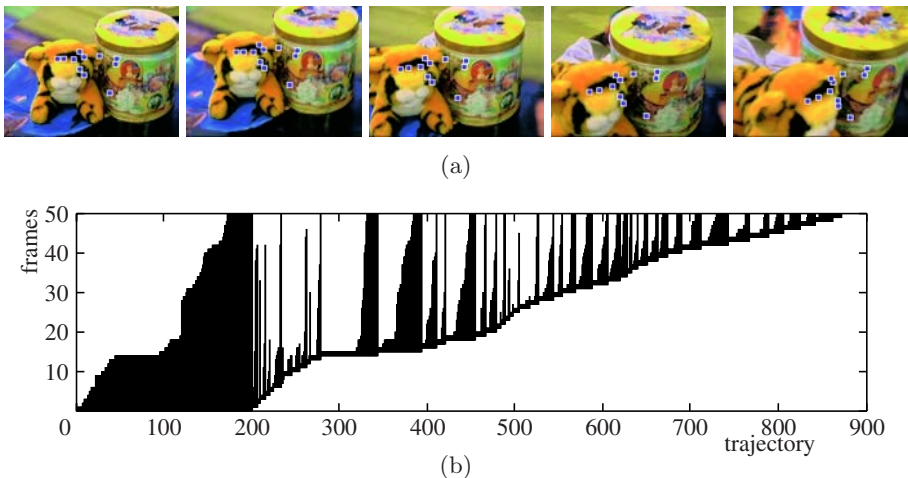


Fig. 2. (a) Five decimated frames from a 50 frame sequence and 11 points correctly tracked throughout the sequence. (b) The life spans of the detected 871 trajectories.

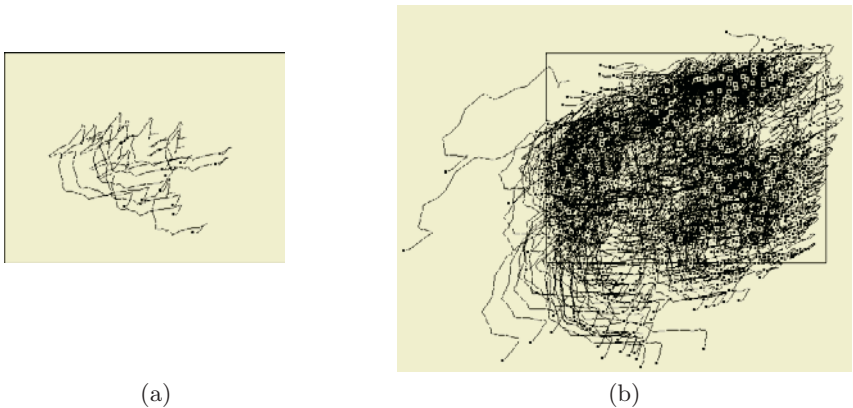


Fig. 3. (a) The 11 complete inlier trajectories. (b) The 560 optimal extensions of the trajectories.

Among them, 29 are complete trajectories, of which 11 are regarded as inliers by the procedure described in Sec. 3. The marks \square in Fig. 2(a) indicate their positions; Fig. 3(a) shows their trajectories.

Using the affine space they define, we extended the partial trajectories and optimized the affine space and the extended trajectories. The optimization converged after 11 iterations, resulting in the 560 inlier trajectories shown in Fig. 3(b). The computation time for this optimization was 134 seconds. We used Pentium 4 2.4GHz for the CPU with 1GB main memory and Linux for the OS.

Fig. 4 shows four enlarged trajectories that underwent significant corrections by the optimization: the trajectories in Fig. 4(a), which appeared to scatter inconsistently, were corrected into those in Fig. 4(b), which are more consistent

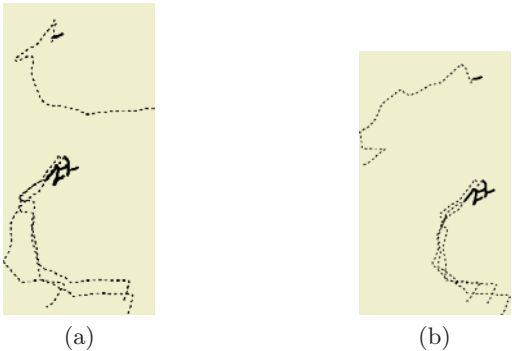


Fig. 4. (a) Four trajectories before optimization. The real lines show the original data, and the dotted lines show the estimated parts. (b) The corresponding optimized trajectories.

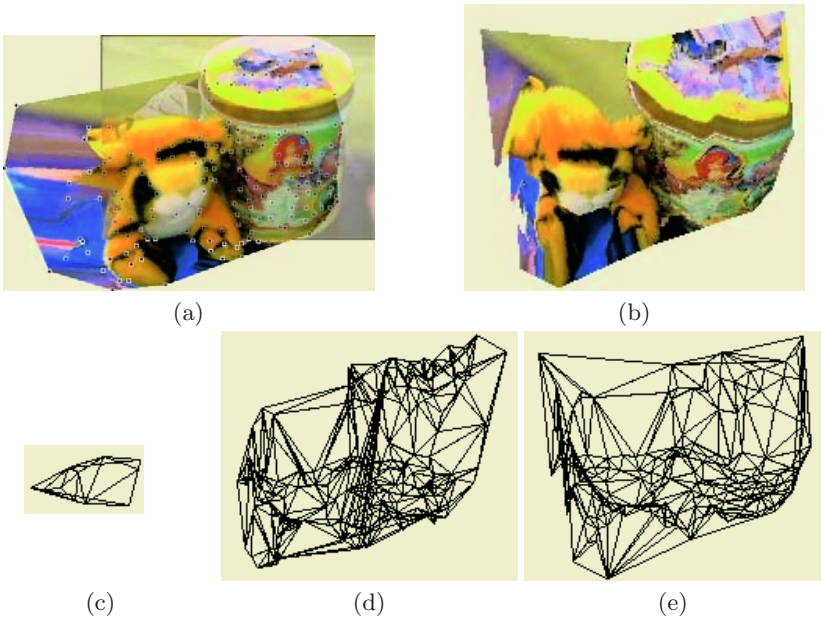


Fig. 5. (a) The extrapolated texture-mapped image of the 33th frame. (b) The reconstructed 3-D shape. (c) The patches reconstructed from the 11 initial complete (d) The patches reconstructed from all extended trajectories without optimization. (e) The corresponding result with optimization.

with the global motion. The solid lines indicate the original data; the dashed lines indicate the estimated parts.

Fig. 5(a) is the extrapolated image of the 33th frame after missing feature positions are restored: using the 180 feature points visible in the first frame, we defined triangular patches, to which the texture in the first frame is mapped.

We reconstructed the 3-D shape by factorization based on weak perspective projection [10]. Fig. 5(b) is the top view of the texture-mapped shape. Fig. 5(c) shows the patches reconstructed from the 11 initial trajectories in Fig. 2(c). Evidently, a meaningful 3-D shape cannot be reconstructed from such a small number of feature points. Fig. 5(d) shows the patches reconstructed from extended trajectories without optimization; Fig. 5(e) is the corresponding shape after optimization.

From these results, we can see that a sufficient number of trajectories can be restored for detailed 3-D reconstruction by extending the partial trajectories and that incorrect trajectories are removed or corrected by the optimization process. According to visual inspection, the reconstructed 3-D shape appears to be better after the optimization, but the difference is small. This is probably because the effects of trajectory errors are suppressed by the factorization algorithm [10], which optimizes the solution using all the data in all the frames.

6 Concluding Remarks

We have presented a new method for extending interrupted feature point tracking for 3-D affine reconstruction. Our method consists of iterations for optimally extending the trajectories and for optimally estimating the affine space. In every step, the reliability of the extended trajectories is tested, and those judged to be outliers are removed. Using real video images, we have demonstrated that a sufficient number of trajectories can be restored for detailed 3-D reconstruction.

Acknowledgments. This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under a Grant in Aid for Scientific Research C(2) (No. 15500113), the Support Center for Advanced Telecommunications Technology Research, and Kayamori Foundation of Informational Science Advancement.

References

1. S. Brandt, Closed-form solutions for affine reconstruction under missing data, *Proc. Statistical Methods in Video Processing Workshop*, Copenhagen, Denmark, June, 2002, pp. 109–114.
2. M. A. Fischer and R. C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. ACM*, **24**-6 (1981-6), 381–395.
3. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, U.K., 2000.
4. D. Q. Huynh and A. Heyden, Outlier detection in video sequences under affine projection, *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, Vol. 2, Kauai, HI, U.S.A., December 2001, pp. 695–701.
5. D. W. Jacobs, Linear fitting with missing data for structure-from-motion, *Comput. Vision Image Understand.*, **82**-1 (2001-4), 57–81.

6. F. Kahl and A. Heyden, Affine structure and motion from points, lines and conics, *Int. J. Comput. Vision*, **33-3** (1999-9), 163–180.
7. K. Kanatani, Motion segmentation by subspace separation and model selection, *Proc. 8th Int. Conf. Comput. Vision*, Vol. 2, Vancouver, Canada, July 2001, pp. 301–306.
8. K. Kanatani, Motion segmentation by subspace separation: Model selection and reliability evaluation, *Int. J. Image Graphics*, **2-2** (2002-4), 179–197.
9. K. Kanatani, Evaluation and selection of models for motion segmentation, *Proc. 7th Euro. Conf. Comput. Vision*, Copenhagen, Denmark, June 2002, pp. 335–349.
10. C. J. Poelman and T. Kanade, A paraperspective factorization method for shape and motion recovery, *IEEE Trans. Patt. Anal. Mach. Intell.*, **19-3** (1997-3), 206–218.
11. P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, Wiley, New York, 1987.
12. H. Saito and S. Kamijima, Factorization method using interpolated feature tracking via projective geometry, *Proc. 14th British Machine Vision Conf.*, Vol. 2, Norwich, UK, September 2003, pp. 449–458.
13. H.-Y. Shum, K. Ikeuchi and R. Reddy, Principal component analysis with missing data and its application to polyhedral object modeling, *IEEE Trans. Patt. Anal. Mach. Intell.*, **17-3** (1995-9), 854–867.
14. Y. Sugaya and K. Kanatani, Outlier removal for motion tracking by subspace separation, *IEICE Trans. Inf. Syst.*, **E86-D-6** (2003-6), 1095–1102.
15. C. Tomasi and T. Kanade, Shape and motion from image streams under orthography—A factorization method, *Int. J. Comput. Vision*, **9-2** (1992-11), 137–154.
16. C. Tomasi and T. Kanade, *Detection and Tracking of Point Features*, CMU Tech. Rep. CMU-CS-91-132, April 1991: <http://vision.stanford.edu/~birch/klt/>.

Many-to-Many Feature Matching Using Spherical Coding of Directed Graphs

M. Fatih Demirci¹, Ali Shokoufandeh¹, Sven Dickinson², Yakov Keselman³,
and Lars Bretzner⁴

¹ Department of Computer Science, Drexel University,
Philadelphia, PA 19104, USA
{mdemirci, ashokouf}@mcs.drexel.edu

² Department of Computer Science, University of Toronto,
Toronto, Ontario, Canada M5S 3G4
sven@cs.toronto.edu

³ School of Computer Science, Telecommunications and Information Systems,
DePaul University, Chicago, IL 60604, USA
ykeselman@cs.depaul.edu

⁴ Computational Vision and Active Perception Laboratory,
Department Of Numerical Analysis and Computer Science,
KTH, Stockholm, Sweden
bretzner@nada.kth.se

Abstract. In recent work, we presented a framework for many-to-many matching of multi-scale feature hierarchies, in which features and their relations were captured in a vertex-labeled, edge-weighted directed graph. The algorithm was based on a metric-tree representation of labeled graphs and their metric embedding into normed vector spaces, using the embedding algorithm of Matoušek [13]. However, the method was limited by the fact that two graphs to be matched were typically embedded into vector spaces with different dimensionality. Before the embeddings could be matched, a dimensionality reduction technique (PCA) was required, which was both costly and prone to error. In this paper, we introduce a more efficient embedding procedure based on a spherical coding of directed graphs. The advantage of this novel embedding technique is that it prescribes a single vector space into which both graphs are embedded. This reduces the problem of directed graph matching to the problem of geometric point matching, for which efficient many-to-many matching algorithms exist, such as the Earth Mover's Distance. We apply the approach to the problem of multi-scale, view-based object recognition, in which an image is decomposed into a set of blobs and ridges with automatic scale selection.

1 Introduction

The problem of object recognition is often formulated as that of matching configurations of image features to configurations of model features. Such configurations are often represented as vertex-labeled graphs, whose nodes represent

image features (or their abstractions), and whose edges represent relations (or constraints) between the features. For scale-space structures, represented as directed graphs, relations can represent both parent/child relations as well as sibling relations. To match two graph representations (hierarchical or otherwise) means to establish correspondences between their nodes. To evaluate the quality of a match, an overall distance measure is defined, whose value depends on both node and edge similarity.

Previous work on graph matching has typically focused on the problem of finding a one-to-one correspondence between the vertices of two graphs. However, the assumption of one-to-one correspondence is a very restrictive one, for it assumes that the primitive features (nodes) in the two graphs agree in their level of abstraction. Unfortunately, there are a variety of conditions that may lead to graphs that represent visually similar image feature configurations yet do not contain a single one-to-one node correspondence.

The limitations of the one-to-one assumption are illustrated in Figure 1, in which an object is decomposed into a set of ridges and blobs extracted at appropriate scales [19]. The ridges and blobs map to nodes in a directed graph, with parent/child edges directed from coarser scale nodes to overlapping finer scale nodes, and sibling edges between nodes that share a parent. Although the two images clearly contain the same object, the decompositions are not identical. Specifically, the ends of the fingers in the right hand have been over-segmented with respect to the left hand. It is quite common that due to noise or segmentation errors, a single feature (node) in one graph can correspond to a collection of broken features (nodes) in another graph. Or, due to scale differences, a single, coarse-grained feature in one graph can correspond to a collection of fine-grained features in another graph. Hence, we seek not a one-to-one correspondence between image features (nodes), but rather a many-to-many correspondence.

In recent work [10,7], we presented a framework for many-to-many matching of undirected graphs and directed graphs, respectively, where features and their relations were represented using edge-weighted graphs. The method began with transforming a graph into a metric tree. Next, using the graph embedding technique of Matoušek [13], the tree was embedded into a normed vector space. This two-step transformation allowed us to reduce the problem of many-to-many graph matching to a much simpler problem of matching weighted distributions of points in a normed vector space. To compute the distance between two weighted distributions, we used a *distribution-based* similarity measure, known as the Earth Mover’s Distance under transformation.

The previous procedure suffered from a significant limitation. Namely, each graph was embedded into a vector space of arbitrary dimensions, and before the embeddings could be matched, a dimensionality reduction step was required, which was both costly and prone to error. Specifically, we used an inefficient Principal Components Analysis (PCA)-based method to project the two distributions into the same normed space. In this paper, we present an entirely different embedding method based on a spherical coding algorithm. This efficient (linear-time) method embeds metric trees into vector spaces of prescribed dimen-

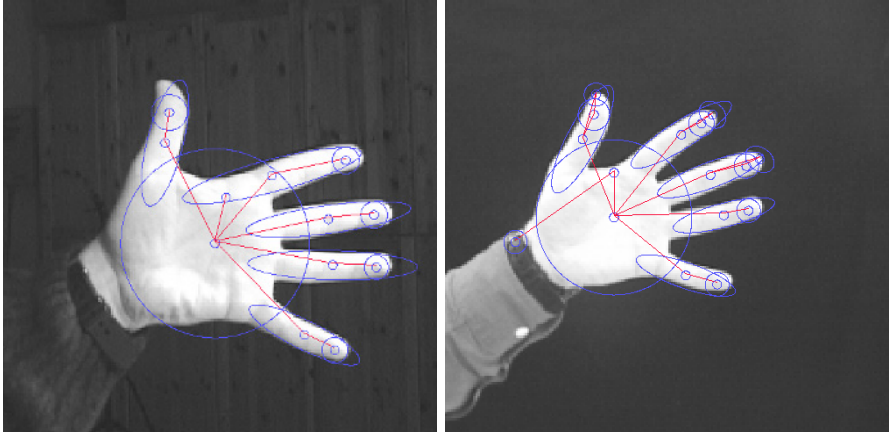


Fig. 1. The Need for Many-to-Many Matching. In the two images, the two objects are similar, but the extracted features are not necessarily one-to-one. Specifically, the ends of the fingers in the left hand have been over-segmented in the right hand.

sionality, precluding the need for a dimensionality reduction step. We demonstrate the framework on the problem of multi-scale shape matching, in which an image is decomposed into a set of blobs and ridges with automatic scale selection.

2 Related Work

The problem of many-to-many graph matching has been studied most often in the context of edit-distance (see, e.g., [14,12,15,18]). In such a setting, one seeks a minimal set of re-labelings, additions, deletions, merges, and splits of nodes and edges that transform one graph into another. However, the edit-distance approach has its drawbacks: 1) it is computationally expensive (p-time algorithms are available only for trees); 2) the method, in its current form, does not accommodate edge weights; 3) the method does not deal well with occlusion and scene clutter, resulting in much effort spent in “editing out” extraneous graph structure; and 4) the cost of an editing operation often fails to reflect the underlying visual information (for example, the visual similarity of a contour and its corresponding broken fragments should not be penalized by the high cost of merging the many fragments).

In the context of line and segment matching, Beveridge and Riseman [3] addressed this problem via exhaustive local search. Although their method found good matches reliably and efficiently (due to their choice of the objective function and a small neighborhood size), it is unclear how the approach can be generalized to other types of feature graphs and objective functions.

In a novel generalization of Scott and Longuet-Higgins [17], Kosinov and Caelli [11] showed how inexact graph matching could be solved using the re-normalization of projections of vertices into the eigenspaces of graphs combined

with a form of relational clustering. Our framework differs from their approach in that: (1) it can handle information encoded in a graph’s nodes, which is desirable in many vision applications; (2) it does not require an explicit clustering step; (3) it provides a well-bounded, low-distortion metric representation of graph structure; (4) it encodes both local and global structure, allowing it to deal with noise and occlusion; and 5) it can accommodate multi-scale representations.

Low-distortion embedding techniques haven proven to be useful in a number of graph algorithms, including clustering and, most recently, on-line algorithms. Indyk [9] provides a comprehensive survey of recent advances and applications of low-distortion graph embedding. Gupta [8] proposes a randomized procedure for embedding metric trees into a vector space of prescribed dimensions. Our spherical coding is a deterministic variation of this procedure. For recent results related to the properties of low-distortion tree embedding, see [1,13].

3 Notation and Definitions

Before describing our many-to-many matching framework, some definitions are in order. To begin, a *graph* G is a pair (\mathcal{A}, E) , where \mathcal{A} is a finite set of vertices and E is a set of connections (edges) between the vertices. An edge $e = (u, v)$ consists of two vertices such that $u, v \in \mathcal{A}$. A graph $G = (\mathcal{A}, E)$ is *edge-weighted*, if each edge $e \in E$ has a weight, $\mathcal{W}(e) \in \mathbb{R}$. Let $G = (\mathcal{A}, E)$ denote an edge-weighted graph with real edge weights $\mathcal{W}(e)$, $e \in E$. We will say that \mathcal{D} is a *metric* for G if, for any three vertices $u, v, w \in \mathcal{A}$, $\mathcal{D}(u, v) = \mathcal{D}(v, u) \geq 0$, with $\mathcal{D}(u, v) = 0$ if and only if $u = v$, and $\mathcal{D}(u, v) \leq \mathcal{D}(u, w) + \mathcal{D}(w, v)$.

One way of defining metric distances on a weighted graph is to use the *shortest-path* metric $\delta(.,.)$ on the graph or its subgraphs, i.e., $\mathcal{D}(u, v) = \delta(u, v)$, the shortest path distance between u and v for all $u, v \in \mathcal{A}$. We will say that the edge weighted tree $\mathfrak{T} = \mathfrak{T}_G(\mathcal{A}', E')$ is a *tree metric* for G , with respect to distance function \mathcal{D} , if for any pair of vertices u, v in G , the length of the unique path between them in \mathfrak{T} is equal to $\mathcal{D}(u, v)$.

An *ultra-metric* is a special type of tree metric defined on rooted trees, where the distance to the root is the same for all leaves in the tree, an approximation that introduces small distortion. A metric \mathcal{D} is an ultra-metric if, for all points x, y, z , we have $\mathcal{D}[x, y] \leq \max\{\mathcal{D}[x, z], \mathcal{D}[y, z]\}$. Unfortunately, an ultra-metric does not satisfy all the properties of a tree metric distance. To create a general tree metric from an ultra-metric, we need to satisfy the *4-point* condition (see [4]): $\mathcal{D}[x, y] + \mathcal{D}[z, w] \leq \max\{\mathcal{D}[x, z] + \mathcal{D}[y, w], \mathcal{D}[x, w] + \mathcal{D}[y, z]\}$, for all x, y, z, w . A metric that satisfies the 4-point condition is called an *additive metric*.

A *metric embedding* is a mapping $f : \mathcal{A} \rightarrow \mathcal{B}$, where \mathcal{A} is a set of points in the original metric space, with distance function $\mathcal{D}(.,.)$, \mathcal{B} is a set of points in the (host) d -dimensional normed space $||.||_k$, and for any pair $p, q \in \mathcal{A}$ we have

$$\frac{1}{c} \mathcal{D}(p, q) \leq ||f(p) - f(q)||_k \leq \mathcal{D}(p, q) \quad (1)$$

for a certain parameter c , known as the *distortion*. Intuitively, such an embedding will enable us to reduce problems defined over *difficult* metric spaces, $(\mathcal{A}, \mathcal{D})$, to

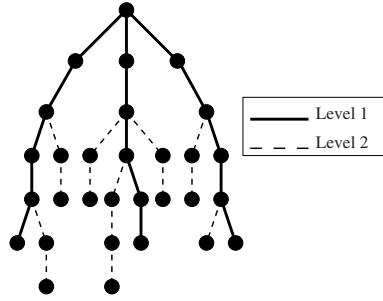


Fig. 2. Path partition of an example tree. The three level 1 paths are bold, while the seven level 2 paths are dashed (there are no other levels in this particular case - see text).

problems over *easier* normed spaces, $(\mathcal{B}, \|\cdot\|_k)$. As can be observed from Equation 1, the distortion parameter c is a critical characteristic of embedding f , i.e., the closer c is to 1, the better the target set \mathcal{B} mimics the original set \mathcal{A} .

To capture the topological structure of a tree, we use the concept of *caterpillar decomposition* and caterpillar dimension. We illustrate the caterpillar decomposition of a rooted tree with no edge weights in Figure 2. The three darkened paths from the root represent three edge-disjoint paths, called level 1 paths. If we remove these three level 1 paths from the tree, we are left with the 7 dashed, edge-disjoint paths. These are the level 2 paths, and if removing them had left additional connected components, the process would be repeated until all the edges in the tree had been removed. The union of the paths is called the caterpillar decomposition, denoted by \mathfrak{P} , and the number of levels in \mathfrak{P} is called the caterpillar dimension, denoted by m .

The caterpillar decomposition \mathfrak{P} can be constructed using a modified depth-first search in linear time. Given a caterpillar decomposition \mathfrak{P} of \mathfrak{T} , we will use L to denote the number of leaves of \mathfrak{T} , and let $P(v)$ represent the unique path between the root and a vertex $v \in \mathcal{A}$. The first segment of $P(v)$ of weight l_1 follows some path P^1 of level 1 in \mathfrak{P} , the second segment of weight l_2 follows a path P^2 of level 2, and the last segment of weight l_α follows a path P^α of level $\alpha \leq m$. The sequences $\langle P^1, \dots, P^\alpha \rangle$ and $\langle l_1, \dots, l_\alpha \rangle$ will be referred to as the *decomposition sequence* and the *weight sequence* of $P(v)$, respectively.

Finally, we introduce the notion of *spherical codes* in our embedding procedure. A spherical code is the distribution of a finite set of n points on the surface of a unit sphere such that the minimum distance between any pair of points is maximized [6]. Equivalently, one can try to minimize the radius r of a d -dimensional sphere such that n points can be placed on the surface, where any two of the points are at angular distance 2 from each other. Recall that the angular distance between two points is the acute angle subtended by them at the origin.

4 Metric Embedding of Graphs via Spherical Coding

4.1 Problem Formulation

Our interest in low-distortion embedding is motivated by its ability to transform the problem of many-to-many matching in finite graphs to the problem of geometric point matching in low-dimensional vector spaces. For graphs, the problem of low-distortion embedding is a challenging one. Let $G_1 = (\mathcal{A}_1, E_1, \mathcal{D}_1)$, $G_2 = (\mathcal{A}_2, E_2, \mathcal{D}_2)$ denote two graphs on vertex sets \mathcal{A}_1 and \mathcal{A}_2 , edge sets E_1 and E_2 , under distance metrics \mathcal{D}_1 and \mathcal{D}_2 , respectively (\mathcal{D}_i represents the distances between all pairs of nodes in G_i). Ideally, we seek a single embedding mechanism that can map each graph to the same vector space, in which the two embeddings can be directly compared.

We will tackle the problem in two steps. Given a d -dimensional target space \mathbb{R}^d , we will seek low-distortion embeddings f_i that map sets \mathcal{A}_i to sets \mathcal{B}_i under distance function $\|\cdot\|_k$, $i \in \{1, 2\}$. The fixed-dimension embedding is based on a novel spherical coding of the shortest-path metric on a tree. To apply this embedding to our directed acyclic graphs therefore requires that we map them to trees with low distortion. It is here that we introduce the concept of relative scale to the points, allowing us to match hierarchical graphs. Using these mappings, the problem of many-to-many hierarchical vertex matching between G_1 and G_2 is reduced to that of computing a mapping \mathcal{M} between subsets of \mathcal{B}_1 and \mathcal{B}_2 .

It is known that a minimum-distortion embedding of a metric tree into the d -dimensional Euclidean space will have distortion of $O(L^{\frac{1}{d-1}} \sqrt{\min(\log L, d)})$, where L is the number of leaves in the tree [8]. Observe that as the dimension d of the target space decreases, the distortion of the embedding increases. We would therefore like to strike a good balance between distortion and dimension.

4.2 Construction of a Tree Metric for a Distance Function

Let $G = (\mathcal{A}, E)$ denote an edge-weighted graph and \mathcal{D} denote a shortest-path metric for G , i.e., $\mathcal{D}(u, v) = \delta(u, v)$, for all $u, v \in \mathcal{A}$. The problem of approximating (or fitting) an $n \times n$ distance matrix \mathcal{D} by a tree metric \mathfrak{T} is known as the *Numerical Taxonomy* problem. Since the numerical taxonomy problem is an open problem for general distance metrics, we must explore approximation methods.

The numerical taxonomy problem can be approximated by converting the distance matrix \mathcal{D} to the weaker ultra-metric distance matrix. To create a general tree metric from an ultra-metric, we need to satisfy the *4-point* condition. Observe that a metric \mathcal{D} is additive if and only if it is a tree metric (see [4]). Therefore, our construction of a tree metric will consist of: 1) constructing an ultra-metric from \mathcal{D} , and 2) modifying the ultra-metric to satisfy the 4-point condition. For details of one such approximation framework, see Agarwala et al. [1]. The construction of a tree metric in their algorithm is achieved by transforming the general tree metric problem to that of ultra-metrics. Their algorithm, which follows the two-step procedure outlined above, generates an approximation (tree

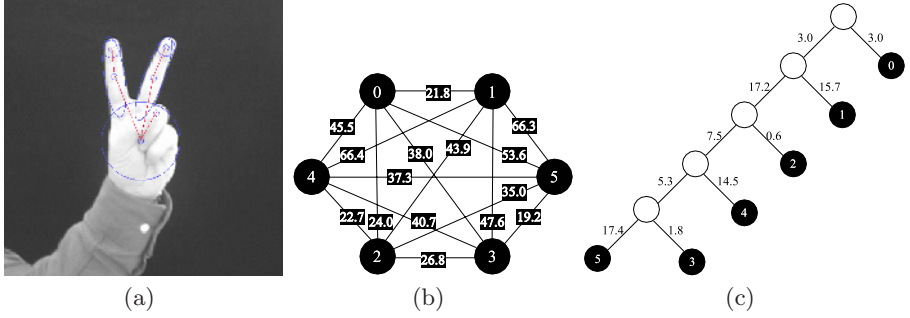


Fig. 3. Metric tree representation of the Euclidean distances between nodes in a graph. The gesture image (a) consists of 6 regions (the region representing the entire hand is not shown). The complete graph in (b) captures the Euclidean distances between the centroids of the regions, while (c) is the metric tree representation of the multi-scale decomposition (with additional vertices).

metric \mathfrak{T}) to an optimal additive metric in time $O(n^2)$. It should be noted that this construction does not necessarily maintain the vertex set of G invariant. We will have to make sure that in the embedding process (see Section 4), the extra vertices generated during the metric tree construction are eliminated. An example of constructing a metric tree from a graph is shown Figure 3.

4.3 Construction of Spherical Codes

To embed our metric trees into Euclidean spaces of fixed dimension, we introduce the concept of *spherical codes*. Such codes, in turn, will allow us to directly compare two embeddings. The embedding framework is best illustrated through an example, in which a weighted tree is embedded into \mathbb{R}^2 , as shown in Figure 4. To ease visualization, we will limit the discussion to the first quadrant. The weighted tree contains 4 paths $\langle a, b, c \rangle$, $\langle a, d, f, h \rangle$, $\langle d, e \rangle$, and $\langle f, g \rangle$ in its caterpillar decomposition. In the embedding, the root is assigned to the origin. Next, we seek a set of 4 vectors, one for each path in the caterpillar decomposition, such that their inner products are minimized, i.e., their endpoints are maximally apart. These vectors define the general directions in which the vertices on each path in the caterpillar decomposition will be embedded.

Three of the four vectors will be used by the caterpillar paths belonging to the subtree rooted at vertex d , and one vector will be used by the path belonging to the subtree rooted at vertex b . This effectively subdivides the first quadrant into two cones, C_b and C_d . The volume of these cones is a function of the number of caterpillar paths belonging to the subtrees rooted at b and d . The cone C_d , in turn, will be divided into two smaller cones, C_e and C_f , corresponding to the subtrees rooted at e and f , respectively. The extreme rays of sub-cones C_b , C_e , and C_f will correspond to the 4 directions defining the embedding. Finally, to complete the embedding, we translate the sub-cones away from the origin along their directional rays to positions defined by the path lengths in the tree. For

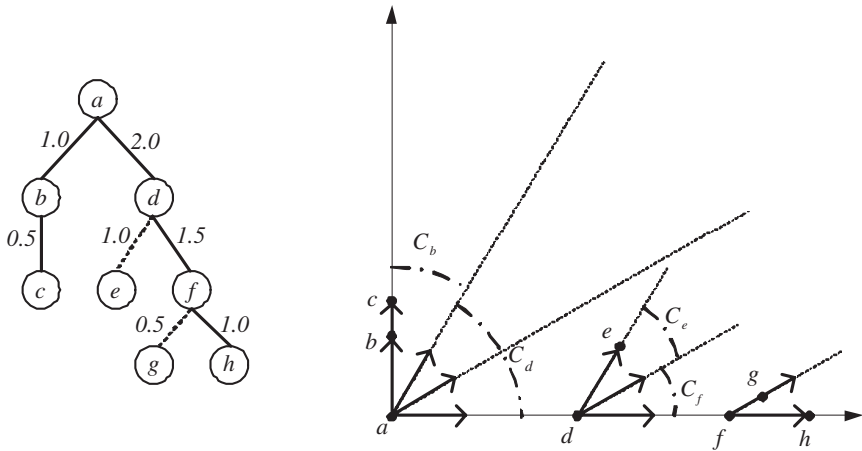


Fig. 4. An edge weighted tree and its spherical code in 2D. The Cartesian coordinates of the vertices are: $a = (0, 0)$, $b = (0, 1.0)$, $c = (0, 1.5)$, $d = (2.0, 0)$, $e = (2.5, 0.87)$, $f = (3.5, 0)$, $g = (3.93, 0.25)$, and $h = (4.5, 0)$.

example, to embed point b , we will move along the extremal ray of C_b and will embed b at $(0, 1.0)$. Similarly, the sub-cone C_d will be translated along the other extremal ray, embedding d at $(2.0, 0)$.

In d -dimensional Euclidean space \mathbb{R}^d , computing the embedding $f : \mathcal{A} \rightarrow \mathcal{B}$ under $\|\cdot\|_2$ is more involved. Let L denote the number of paths in the caterpillar decomposition. The embedding procedure defines L vectors in \mathbb{R}^d that have a large angle with respect to each other on the surface of a hypersphere S_d of radius r . These vectors are chosen in such a way that any two of their endpoints on the surface \sum_d are at least spherical distance 2 from each other. We will refer to such vectors as *well-separated*. Consider the set of hyperplanes $H_i = (0, 2, 4, \dots, 2i)$, and let $\sum_d(i) = H_i \cap \sum_d$. Since each of the $\sum_d(i)$ are hypercircles, i.e., surfaces of spheres in dimension $d-1$, we can recursively construct well-separated vectors on each hypercircle $\sum_d(i)$. Our construction stops when the sphere becomes a circle and the surface becomes a point in 2 dimensions. It is known that taking r to be $O(dL^{1/d-1})$, and the minimum angle between two vectors to be $2/r$ provides us with L well-separated vectors [6]. In Figure 4, we have 4 such vectors emanating from the origin.

Now that the embedding directions have been established, we can proceed with the embedding of the vertices. The embedding procedure starts from the root (always embedded at the origin) and embeds vertices following the embedding of their parents. For each vertex in the metric tree \mathfrak{T} , we associate with every subtree \mathfrak{T}_v a set of vectors C_v , such that the number of vectors in C_v equals the number of paths in the caterpillar decomposition of \mathfrak{T}_v . Initially, the root has the entire set of L vectors. Consider a subtree rooted at vertex v , and let us assume that vertex v has k children, v_1, \dots, v_k . We partition the set of vectors into k subsets, such that the number of vectors in each subset, S_v , equals

the number of leaves in \mathfrak{T}_v . We then embed the vertex v_l ($1 \leq l \leq k$) at the position $f(v) + w_l * x_l$, where w_l is the length of the edge (v, v_l) and x_l is some vector in C_v . We recursively repeat the same process for each subtree rooted at every child of v , and stop when there are no more subtrees to consider.

4.4 Encoding Directed Edges

The distance metric defined on the graph structure is based on the undirected edge weights. While the above embedding has preserved the distance metric, it has failed to preserve any oriented relations, such as the hierarchical relations common to scale-space structures. This is due to the fact that oriented relations do not satisfy the symmetry property of a metric. We can retain this important information in our embedding by moving it into the nodes as node attributes, a technique used in the encoding of directed topological structure in [20], directed geometric structure in [19], and shape context in [2]. Encoding in a node the attributes of the oriented edges incident to the node requires computing distributions on the attributes and assigning them to the node. For example, a node with a single parent at a coarser scale and two children at a finer scale might encode a relative scale distribution (histogram) as a node attribute. The resulting attribute provides a contextual signature for the node which will be used by the matcher (Section 5) to reduce matching ambiguity.

Specifically, let $G = (\mathcal{A}, E)$ be a graph to be embedded. For every pair of vertices, (u, v) , we let $R_{u,v}$ denote the attribute vector associated with the pair. The entries of each such vector represent the set of oriented relations R between u, v . For a vertex $u \in \mathcal{A}$, we let $N(u)$ denote the set of vertices $v \in \mathcal{A}$ adjacent to u . For a relation $p \in R$, we will denote $\mathcal{P}(u, p)$ as the set of values for relation p between u and all vertices in $N(u)$, i.e., $\mathcal{P}(u, p)$ corresponds to entry p of vector $R_{u,v}$ for $v \in N(u)$. Feature vector \mathcal{P}_u for point u is the set of all $\mathcal{P}(u, p)$'s for $p \in R$. Observe that every entry $\mathcal{P}(u, p)$ of vector \mathcal{P}_u can be considered as a local distribution (*histogram*) of feature p in the neighborhood $N(u)$ of u . We adopt the method of [19], in which the distance function for two such vectors \mathcal{P}_u and \mathcal{P}_p is computed through a weighted combination of Hausdorff distances between $\mathcal{P}(u, p)$ and $\mathcal{P}(u', p)$ for all values of p .

5 Distribution-Based Many-to-Many Matching

By embedding vertex-labeled graphs into normed spaces, we have reduced the problem of many-to-many matching of graphs to that of many-to-many matching of weighted distributions of points in normed spaces. Given a pair of weighted distributions in the same normed space, the Earth Mover's Distance (EMD) framework [16] is then applied to find an optimal match between the distributions. The EMD approach computes the minimum amount of work (defined in terms of displacements of the masses associated with points) it takes to transform one distribution into another. The EMD approach assumes that a distance measure between single features, called the *ground distance*, is given. The EMD

then “lifts” this distance from individual features to full distributions. The main advantage of using EMD lies in the fact that it subsumes many histogram distances and permits partial matches in a natural way. This important property allows the similarity measure to deal with uneven clusters and noisy datasets. Details of the method, along with an extension, are presented in [10].

The standard EMD formulation assumes that the two distributions have been aligned. However, recall that a translated and rotated version of a graph embedding will also be a graph embedding. To accommodate pairs of distributions that are “not rigidly embedded”, Cohen and Guibas [5] extended the definition of EMD, originally applicable to pairs of fixed sets of points, to allow one of the sets to undergo a transformation. They also suggested an iterative process (which they call **FT**, short for “an optimal **F**low and an optimal **T**ransformation”) that achieves a local minimum of the objective function. Details on how we compute the optimal transformation can be found in [10,7].

5.1 The Final Algorithm

Our algorithm for many-to-many matching is a combination of the previous procedures, and is summarized as follows:

Algorithm 1 Many-to-many graph matching

- 1: Compute the metric tree \mathfrak{T}_i corresponding to G_i according to Section 4 (see [1] for details).
 - 2: Construct low-distortion embeddings $f_i(\mathfrak{T}_i)$ of \mathfrak{T}_i into $(\mathcal{B}_i, ||\cdot||_2)$ according to Section 4.
 - 3: Compute the EMD between \mathcal{E}_i ’s by applying the FT iteration, computing the optimal transformation T according to Section 5 (see [10] for details).
 - 4: Interpret the resulting optimal flow between \mathcal{E}_i ’s as a many-to-many vertex matching between G_i ’s.
-

6 Experiments

As an illustration of our approach, let’s first return to the example shown in Figure 1, where we observed the need for many-to-many matching. The results of applying our method to these two images is shown in Figure 5, in which many-to-many feature correspondences have been colored the same. For example, a set of blobs and ridges describing a finger in the left image is mapped to a set of blobs in ridges on the corresponding finger in the right image.

To provide a more comprehensive evaluation, we tested our framework on two separate image libraries, the Columbia University COIL-20 (20 objects, 72 views per object) and the ETH Zurich ETH-80 (8 categories, 10 exemplars per

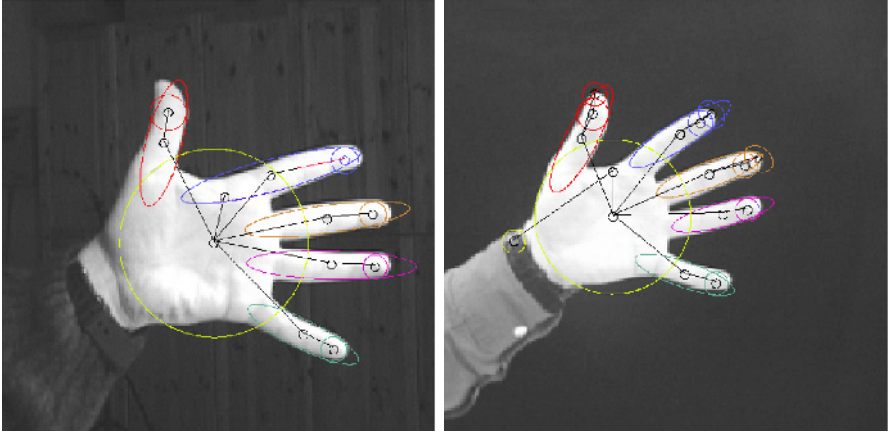


Fig. 5. Applying our algorithm to the images in Figure 1. Many-to-many feature correspondences have been colored the same.

category, 41 views per exemplar)¹. For each view, we compute a multi-scale blob decomposition, using the algorithm described in [19]. Next, we compute the tree metric corresponding to the complete edge-weighted graph defined on the regions of the scale-space decomposition of the view. The edge weights are computed as a function of the distances between the centroids of the regions in the scale-space representation. Finally, each tree is embedded into a normed space of prescribed dimension. This procedure results in two databases of weighted point sets, each point set representing an embedded graph.

For the COIL-20 database, we begin by removing 36 (of the 72) representative views of each object (every other view), and use these removed views as queries to the remaining view database (the other 36 views for each of the 20 objects). We then compute the distance between each “query” view and each of the remaining database views, using our proposed matching algorithm. Ideally, for any given query view i of object j , $v_{i,j}$, the matching algorithm should return either $v_{i+1,j}$ or $v_{i-1,j}$ as the closest view. We will classify this as a correct matching. Based on the overall matching statistics, we observe that in all but 4.8% of the experiments, the closest match selected by our algorithm was a neighboring view. Moreover, among the mismatches, the closest view belonged to the same object in 81.02% of the cases. In comparison to the many-to-many

¹ Arguably, the COIL database is not the ideal testbed for an image representation (in our case, a multi-scale blob and ridge decomposition) whose goal is to describe the coarse shape of an object. Unlike the PCA-based image characterization for which the COIL database was originally created, the multi-scale blob and ridge decomposition provides invariance to translation, rotation, scale, minor part deformation and articulation, and minor within-class shape deformation. Although a standard database for recognition testing, the COIL database does not exercise these invariants.

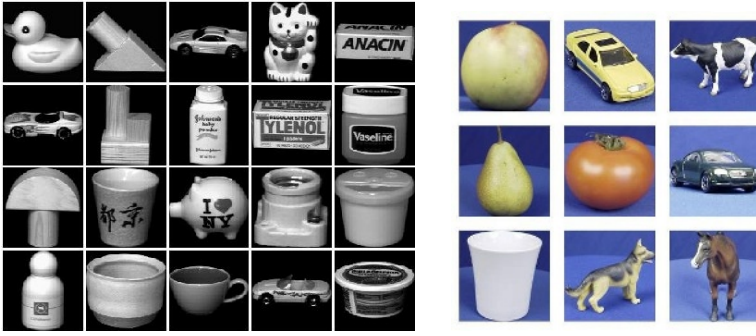


Fig. 6. Views of sample objects from the Columbia University Image Library (COIL-20) and the ETH Zurich (ETH-80) Image Set.

matching algorithm based on PCA embedding [7] for a similar setup, the new procedure showed an improvement of 5.5%.

It should be pointed out that these results can be considered worst case for two reasons. First, the original 72 views per object sampling resolution was tuned for an eigenimage approach. Given the high similarity among neighboring views, it could be argued that our matching criterion is overly harsh, and that perhaps a measure of “viewpoint distance”, i.e., “how many views away was the closest match” would be less severe. In any case, we anticipate that with fewer samples per object, neighboring views would be more dissimilar, and our matching results would improve. Second, and perhaps more importantly, many of the objects are symmetric, and if a query neighbor has an identical view elsewhere on the object, that view might be chosen (with equal distance) and scored as an error. Many of the objects in the database are rotationally symmetric, yielding identical views from each viewpoint.

For the ETH-80 database, we chose a subset of 32 objects (4 from each of the 8 categories) with full sampling (41 views) per object. For each object, we removed each of its 41 views from the database, one view at a time, and used the removed view as a query to the remaining view database. We then computed the distance between each query view and each of the remaining database views. The criteria for correct classification was similar to the COIL-20 experiment. Our experiments showed that in all but 6.2% of the experiments, the closest match selected by our algorithm was a neighboring view. Among the mismatches, the closest view belonged to the same object in 77.19% of the cases, and the same category in 96.27% of the cases. Again, these results can be considered worst case for the same reasons discussed above for the COIL-20 experiment.

Both the embedding and matching procedures can accommodate local perturbation, due to noise and occlusion, because path partitions provide locality. If a portion of the graph is corrupted, the projections of unperturbed nodes will not be affected. Moreover, the matching procedure is an iterative process driven by flow optimization which, in turn, depends only on local features, and is thereby

Table 1. Recognition rate as a function of increasing perturbation. Note that the baseline recognition rate (with no perturbation) is 95.2%

PERTURBATION	5%	10%	15%	20%
RECOGNITION RATE	91.07%	88.13%	83.68%	77.72%

unaffected by local perturbation. To demonstrate the framework’s robustness, we performed four perturbation experiments on the COIL-20 database. The experiments are identical to the COIL-20 experiment above, except that the query graph was perturbed by adding/deleting 5%, 10%, 15%, and 20% of its nodes (and their adjoining edges). The results are shown in Table 1, and reveal that the error rate increases gracefully as a function of increased perturbation.

7 Conclusions

We have presented a novel, computationally efficient approach to many-to-many matching of directed graphs. To match two graphs, we begin by constructing metric tree representations of the graphs. Next, we embed them in a geometric space with low distortion using a novel encoding of the graph’s vertices, called spherical codes. Many-to-many graph matching now becomes a many-to-many geometric point matching problem, for which the Earth Mover’s Distance algorithm is ideally suited. Moreover, by mapping a node’s geometric and structural “context” in the graph to an attribute vector assigned to its corresponding point, we can extend the technique to deal with hierarchical graphs that represent multi-scale structure. We evaluate the technique on two major image databases, using a multi-scale image representation that captures coarse image structure, and include a set of structural perturbation experiments to show the algorithm’s robustness to graph “noise”.

Acknowledgment. Ali Shokoufandeh acknowledges the partial support provided by grants from the National Science Foundation and the Office of Naval Research. The work of Yakov Keselman is supported, in part, by the NSF grant No. 0125068. Sven Dickinson acknowledges the support of NSERC, CITO, IRIS, PREA, and the NSF. The authors would also like to thank Shree Nayar and Bernt Schiele for their COIL-20 and ETH-80 databases, respectively.

References

1. R. Agarwala, V. Bafna, M. Farach, M. Paterson, and M. Thorup. On the approximability of numerical taxonomy (fitting distances by tree metrics). *SIAM Journal on Computing*, 28(2):1073–1085, 1999.
2. S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE PAMI*, 24(4):509–522, April 2002.

3. R. Beveridge and E. M. Riseman. How easy is matching 2D line models using local search? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):564–579, June 1997.
4. P. Buneman. The recovery of trees from measures of dissimilarity. In F. Hodson, D. Kendall, and P. Tautu, editors, *Mathematics in the Archaeological and Historical Sciences*, pages 387–395. Edinburgh University Press, Edinburgh, 1971.
5. S. D. Cohen and L. J. Guibas. The earth mover’s distance under transformation sets. In *Proceedings, 7th International Conference on Computer Vision*, pages 1076–1083, Kerkyra, Greece, 1999.
6. J. H. Conway and N. J. A. Sloane. *Sphere Packing, Lattices and Groups*. Springer-Verlag, New York, 1998.
7. F. Demirci, A. Shokoufandeh, Y. Keselman, S. Dickinson, and L. Bretzner. Many-to-many matching of scale-space feature hierarchies using metric embedding. In *Scale Space Methods in Computer Vision, 4th International Conference*, pages 17–32, Isle of Skye, UK, June, 10–12 2003.
8. A. Gupta. Embedding tree metrics into low dimensional euclidean spaces. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 694–700, 1999.
9. P. Indyk. Algorithmic aspects of geometric embeddings. In *Proceedings, 42nd Annual Symposium on Foundations of Computer Science*, 2001.
10. Y. Keselman, A. Shokoufandeh, F. Demirci, and S. Dickinson. Many-to-many graph matching via low-distortion embedding. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, Madison, WI, June 2003.
11. S. Kosinov and T. Caelli. Inexact multisubgraph matching using graph eigenspace and clustering models. In *Proceedings of SSPR/SPR*, volume 2396, pages 133–142. Springer, 2002.
12. T.-L. Liu and D. Geiger. Approximate tree matching and shape similarity. In *Proceedings, 7th International Conference on Computer Vision*, pages 456–462, Kerkyra, Greece, 1999.
13. J. Matoušek. On embedding trees into uniformly convex Banach spaces. *Israel Journal of Mathematics*, 237:221–237, 1999.
14. B. Messmer and H. Bunke. Efficient error-tolerant subgraph isomorphism detection. In D. Dori and A. Bruckstein, editors, *Shape, Structure and Pattern Recognition*, pages 231–240. World Scientific Publ. Co., 1995.
15. R. Myers, R. Wilson, and E. Hancock. Bayesian graph edit distance. *IEEE PAMI*, 22(6):628–635, 2000.
16. Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.
17. G. Scott and H. Longuet-Higgins. An algorithm for associating the features of two patterns. *Proceedings of Royal Society of London*, B244:21–26, 1991.
18. T. Sebastian, P. Klein, and B. Kimia. Recognition of shapes by editing shock graphs. In *IEEE International Conference on Computer Vision*, pages 755–762, 2001.
19. A. Shokoufandeh, S.J. Dickinson, C. Jönsson, L. Bretzner, and T. Lindeberg. On the representation and matching of qualitative shape at multiple scales. In *Proceedings, 7th European Conference on Computer Vision*, volume 3, pages 759–775, 2002.
20. K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 30:1–24, 1999.

Coupled-Contour Tracking through Non-orthogonal Projections and Fusion for Echocardiography

Xiang Sean Zhou¹, Dorin Comaniciu¹, and Sriram Krishnan²

¹ Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540, USA

² Siemens Medical Solutions, 51 Valley Stream Pkwy, Malvern, PA 19355, USA
xiang.zhou, dorin.comaniciu, sriram.krishnan@siemens.com

Abstract. Existing methods for incorporating subspace model constraints in contour tracking use only partial information from the measurements and model distribution. We propose a complete fusion formulation for robust contour tracking, optimally resolving uncertainties from heteroscedastic measurement noise, system dynamics, and a subspace model. The resulting *non-orthogonal subspace projection* is a natural extension of the traditional model constraint using orthogonal projection. We build models for coupled double-contours, and exploit information from the ground truth initialization through a strong model adaptation. Our framework is applied for tracking in echocardiograms where the noise is *heteroscedastic*, each heart has distinct shape, and the relative motions of epi- and endocardial borders reveal crucial diagnostic features. The proposed method significantly outperforms the traditional shape-space-constrained tracking algorithm. Due to the joint fusion of heteroscedastic uncertainties, the strong model adaptation, and the coupled tracking of double-contours, robust performance is observed even on the most challenging cases.

1 Introduction

Model constraints can significantly improve the performance of a contour tracking algorithm. In most cases, a subspace model is appropriate since the number of modes capturing the major shape variations is limited and usually much smaller than the original number of feature components used to describe the shape [1]. A traditional treatment is to project into a PCA subspace [2,1]. However, this approach does not take advantage of heteroscedastic (i.e., both anisotropic and inhomogeneous) measurement uncertainties [3,4] (See Figure 1). Intuitively, a tracking algorithm should downplay measurements from uncertain regions when consulting a shape model.

A more interesting solution was to directly incorporate a PCA shape space constraint into a Kalman filter-based tracker. In [7,8], the proposal was to set the system noise covariance matrix to be the covariance of a PCA shape model. Nevertheless, this treatment has some limitations. First of all, it did not provide a systematic and complete fusion of the model information because, for example, the model mean is discarded (—as a result, the projection can be arbitrarily far from the model mean in the subspace). Secondly, it mixes the uncertainty from system dynamics with the uncertainty from the statistical shape constraint, while these two can be conceptually different. For example, we may want to use the dynamic model to capture different modes of global rigid motion, while

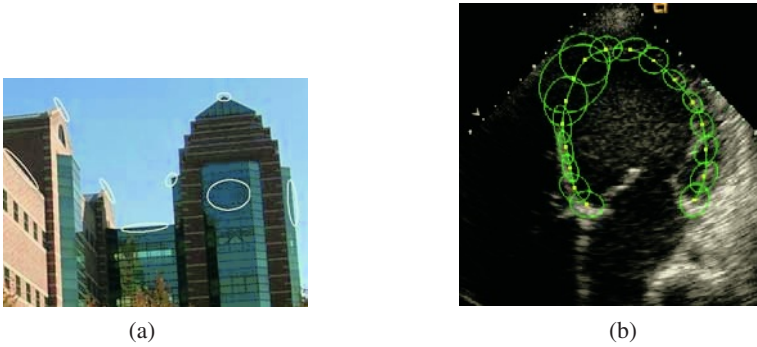


Fig. 1. Ellipses depicting uncertainties in feature localization and motion estimation. The heteroscedastic nature stems from either the *aperture problem* [4,5], or for echocardiograms (b), the *acoustic drop-out* [6].

applying a statistical shape model to control the modes and range of shape variations. Finally, existing solutions do not specifically address the issue of heteroscedastic measurement noise and its influence during the fusion with other information sources. When measurement noise is anisotropic and inhomogeneous, joint fusion of all information sources becomes critical for achieving reliable performance.

We decouple the uncertainty in system dynamics and the statistical shape constraint, and introduce a unified framework for fusing *a subspace shape model with the system dynamics and the measurements with heteroscedastic noise*. We build models for coupled double-contours so that more information can be integrated especially for very noisy data. The double-contour also achieves better preservation of topology¹. To accommodate individual shape characteristics, the generic shape model is strongly adapted using information given about the current case. The subspace model can take the form of a specific subspace distribution, e.g., a Gaussian, or a simple subspace constraint, e.g., the eigenspace model [2,12]. Unlike existing *ad hoc* formulations, our framework treats the two cases in a consistent way, and combines such constraints seamlessly into the tracking framework. The new approach calls for reliable estimation of measurement uncertainties, for which we employ a recent robust solution to the motion estimation problem, which also computes the motion flow uncertainties [13].

The paper is organized as follows: The new model-constrained tracking formulation is presented in Section 2. Section 3 discusses a model adaptation scheme. Section 4 contains experimental evaluation and analysis. Related work and future directions are discussed in Sections 5 and 6, respectively.

2 Model Constraint through Projection and Fusion

Throughout this paper, we represent shapes by control or landmark points, assuming correspondence. These points are fitted by splines before shown to the user. For more implementation details, please refer to Section 4.

¹ Our coupling is probabilistic (governed by the training set) and “soft”(See Section 4.4). For deterministic coupling, “soft” or “hard”, please refer to [9,10,11] and the references therein.

A typical tracking framework fuses information from the dynamic prediction and from noisy measurements. For shape tracking, additional constraints are necessary to stabilize the overall shape in a feasible space/range. In this section, we first extend the traditional subspace constraint using orthogonal projection to non-orthogonal projection. Then, we show that with a complete subspace model constraint, considering also the model mean, this can be further generalized into an information fusion formulation. Finally, these formulas are uniformly combined into a tracking framework.

2.1 Non-orthogonal Projection for Heteroscedastic Noise

Given an n -dimensional measurement point², \mathbf{x} , with uncertainty characterized by a covariance matrix \mathbf{C} , we want to find the “closest” point \mathbf{y}^* in a p -dimensional ($p < n$) subspace, with its axes defined by the orthonormal column vectors of an $n \times p$ matrix, \mathbf{U}_p , $\mathbf{U}_p^T \mathbf{U}_p = \mathbf{I}$, such that the Mahalanobis distance is minimized, i.e., $\mathbf{y}^* = \argmin d^2$, where

$$d^2 = (\mathbf{U}_p \mathbf{y} - \mathbf{x})^T \mathbf{C}^{-1} (\mathbf{U}_p \mathbf{y} - \mathbf{x}). \quad (1)$$

This is in the form of a *weighted least square* ([14], p. 386). By taking derivative of above with respect to \mathbf{y} and setting it to 0, we have

$$\mathbf{y}^* = \mathbf{C}_{\mathbf{y}^*} \mathbf{U}_p^T \mathbf{C}^{-1} \mathbf{x}, \quad \mathbf{C}_{\mathbf{y}^*} = (\mathbf{U}_p^T \mathbf{C}^{-1} \mathbf{U}_p)^{-1}. \quad (2)$$

In general, this is a *non-orthogonal projection*. It is easy to show that the Gaussian $\mathcal{N}(\mathbf{y}^*, \mathbf{C}_{\mathbf{y}^*})$ is the conditional distribution, or *intersection*, of \mathbf{x} in the subspace. Only when $\mathbf{C} = c\mathbf{I}$ with some positive scalar c , we have

$$\mathbf{y}^* = (c^{-1} \mathbf{U}_p^T \mathbf{I} \mathbf{U}_p)^{-1} \mathbf{U}_p^T (c\mathbf{I})^{-1} \mathbf{x} = \mathbf{U}_p^T \mathbf{x}, \quad \mathbf{C}_{\mathbf{y}^*} = c\mathbf{I}_p \quad (3)$$

In our application, this means that all control points on the contour have isotropic and homogeneous uncertainties and the solution reduces to classical orthogonal projection.

2.2 Incorporating Model Distribution through Subspace Fusion

In the above we only considered the subspace constraint while the actual model distribution (assumed Gaussian with mean and covariance) represents important prior information that should not be discarded. In the sequel we show that an information fusion formulation unifies all cases within a general maximal likelihood framework.

The *information space* is the space obtained by multiplying a vector by its corresponding *information matrix*, which is, in the Gaussian case, the inverse of the error covariance matrix. Given two noisy measurements of an n -dimensional variable \mathbf{x} , each with a Gaussian distribution, $\mathcal{N}(\mathbf{x}_1, \mathbf{C}_1)$ and $\mathcal{N}(\mathbf{x}_2, \mathbf{C}_2)$, the maximum likelihood estimate of \mathbf{x} is the point with the minimal sum of Mahalanobis distances, $\mathbf{D}^2(\mathbf{x}, \mathbf{x}_i, \mathbf{C}_i) = (\mathbf{x} - \mathbf{x}_i)^T \mathbf{C}_i^{-1} (\mathbf{x} - \mathbf{x}_i)$, to the two centroids, i.e., $\mathbf{x}^* = \argmin d^2$ with

$$d^2 = (\mathbf{x} - \mathbf{x}_1)^T \mathbf{C}_1^{-1} (\mathbf{x} - \mathbf{x}_1) + (\mathbf{x} - \mathbf{x}_2)^T \mathbf{C}_2^{-1} (\mathbf{x} - \mathbf{x}_2) \quad (4)$$

² Care should be taken to avoid confusion over the interpretation of the term “point”: the *point* here would correspond to a contour with multiple *control points*. By “inhomogeneous” noise, we refer to the inhomogeneity among different *control points* [4].

Taking derivative with respect to \mathbf{x} and setting it to zero, we get:

$$\mathbf{x}^* = \mathbf{C}(\mathbf{C}_1^{-1}\mathbf{x} + \mathbf{C}_2^{-1}\mathbf{x}_2), \quad \mathbf{C} = (\mathbf{C}_1^{-1} + \mathbf{C}_2^{-1})^{-1} \quad (5)$$

which is also known as the best linear unbiased estimate (BLUE) of \mathbf{x} ([15,16]).

When one of the Gaussians is in a subspace of dimension p , e.g., \mathbf{C}_2 is singular, the second term of Eq. (4) can be re-written using pseudoinverse of \mathbf{C}_2 , \mathbf{C}_2^+ :

$$\mathbf{D}^2(\mathbf{x}, \mathbf{x}_2, \mathbf{C}_2) = \sum_{i=1}^p \lambda_i^{-1} [\mathbf{U}_p^T(\mathbf{x} - \mathbf{x}_2)]^2 \equiv (\mathbf{x} - \mathbf{x}_2)^T \mathbf{C}_2^+ (\mathbf{x} - \mathbf{x}_2) \quad (6)$$

with the additional constraint of $\mathbf{U}_0^T \mathbf{x} = 0$ (otherwise, d will diverge). Here $\mathbf{C}_2 = \mathbf{U}\mathbf{A}\mathbf{U}^T$, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n]$, $\mathbf{U}_p = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p]$, $\mathbf{U}_0 = [\mathbf{u}_{p+1}, \mathbf{u}_{p+2}, \dots, \mathbf{u}_n]$, with \mathbf{u}_i 's orthonormal and $\mathbf{A} = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_p, 0, \dots, 0\}$. (Here we have assumed, without loss of generality, that the subspace passes through the origin of the original space.)

With $\mathbf{U}_0^T \mathbf{x} = 0$, \mathbf{x} resides in the subspace as $\mathbf{y} = \mathbf{U}_p^T \mathbf{x}$. Eq. (4) now takes the following general form:

$$d^2 = (\mathbf{U}_p \mathbf{y} - \mathbf{x}_1)^T \mathbf{C}_1^{-1} (\mathbf{U}_p \mathbf{y} - \mathbf{x}_1) + (\mathbf{U}_p \mathbf{y} - \mathbf{x}_2)^T \mathbf{C}_2^+ (\mathbf{U}_p \mathbf{y} - \mathbf{x}_2) \quad (7)$$

Taking derivative with respect to \mathbf{y} yields the fusion estimator for the subspace:

$$\mathbf{y}^* = \mathbf{C}_{\mathbf{y}^*} \mathbf{U}_p^T (\mathbf{C}_1^{-1} \mathbf{x}_1 + \mathbf{C}_2^+ \mathbf{x}_2), \quad \mathbf{C}_{\mathbf{y}^*} = [\mathbf{U}_p^T (\mathbf{C}_1^{-1} + \mathbf{C}_2^+) \mathbf{U}_p]^{-1} \quad (8)$$

Equivalent expressions can be obtained in the original space as:

$$\mathbf{x}^* = \mathbf{U}_p \mathbf{y}^* = \mathbf{C}_{\mathbf{x}^*} (\mathbf{C}_1^{-1} \mathbf{x}_1 + \mathbf{C}_2^+ \mathbf{x}_2), \quad \mathbf{C}_{\mathbf{x}^*} = \mathbf{U}_p \mathbf{C}_{\mathbf{y}^*} \mathbf{U}_p^T \quad (9)$$

It is easy to show that $\mathbf{C}_{\mathbf{x}^*}$ and $\mathbf{C}_{\mathbf{y}^*}$ are the covariance matrices for \mathbf{x}^* and \mathbf{y}^* .

Alternatively, we can write Eq. (8) as

$$\mathbf{y}^* = (\mathbf{U}_p^T \mathbf{C}_1^{-1} \mathbf{U}_p + \mathbf{A}_p^{-1})^{-1} (\mathbf{U}_p^T \mathbf{C}_1^{-1} \mathbf{x}_1 + \mathbf{A}_p^{-1} \mathbf{y}_2) \quad (10)$$

Here $\mathbf{y}_2 = \mathbf{U}_p^T \mathbf{x}_2$, and $\mathbf{A}_p = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_p\}$. Interestingly, Eq. (10) is in fact the BLUE fusion of two subspace Gaussian distributions, one being $\mathcal{N}(\mathbf{y}_2, \mathbf{A}_p)$ and the other being the *non-orthogonal projection* of $\mathcal{N}(\mathbf{x}_1, \mathbf{C}_1)$ in the subspace, $\mathcal{N}((\mathbf{U}_p^T \mathbf{C}_1^{-1} \mathbf{U}_p)^{-1} \mathbf{U}_p^T \mathbf{C}_1^{-1} \mathbf{x}_1, (\mathbf{U}_p^T \mathbf{C}_1^{-1} \mathbf{U}_p)^{-1})$ (cf. Eq. (2)).

2.3 Constrained Tracking through Fusion and Projection

To integrate the above projection and fusion formulas into a tracking framework, we first note that Kalman filter is essentially *fusion* in nature, which is evident in its information filter form ([17], page 138), which :

$$\mathbf{x}_{k+1|k+1} = (\mathbf{P}_{k+1|k}^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} (\mathbf{P}_{k+1|k}^{-1} \mathbf{x}_{k+1|k} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{z}_{k+1}) \quad (11)$$

Here $\mathbf{x}_{i|j}$ is the state estimate at time i given the state or measurement at time j , \mathbf{P} is the state covariance, and \mathbf{H} is the measurement matrix. The measurement model is

$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{r}_k$, where \mathbf{r}_k represents measurement noise with covariance \mathbf{R} . \mathbf{P} is recursively updated as $\mathbf{P}_{k+1|k} = \mathbf{S}\mathbf{P}_{k|k}\mathbf{S}^T + \mathbf{Q}$ using information from a dynamic system model $\mathbf{x}_{k+1} = \mathbf{S}\mathbf{x}_k + \mathbf{q}_k$, where \mathbf{q}_k represents system noise with covariance \mathbf{Q} [17].

For the special case where \mathbf{H} is a square matrix and admits an inverse, we can see Eq. (11) in a strict information fusion form, namely, the fusion of prediction and measurement in the information space (*cf.* Eq. (5)):

$$\mathbf{x}_{k+1|k+1} = (\mathbf{P}_{k+1|k}^{-1} + \mathbf{R}_x^{-1})^{-1} \left[\mathbf{P}_{k+1|k}^{-1} \mathbf{x}_{k+1|k} + \mathbf{R}_x^{-1} \mathbf{x}_{z,k+1} \right] \quad (12)$$

where $\mathbf{R}_x = \mathbf{H}^{-1}\mathbf{R}(\mathbf{H}^{-1})^T$ and $\mathbf{x}_{z,k+1} = \mathbf{H}^{-1}\mathbf{z}_{k+1}$.

Because Kalman filter is a fusion filter and the information fusion operation is *associative*, we can apply the subspace fusion formula, Eq. (8), on the Kalman fusion result of Eq. (11) (In general \mathbf{H} is not invertible; otherwise, Eq. (12) can be used.) and a subspace source $\mathcal{N}(\mathbf{x}_2, \mathbf{C}_2)$, to obtain a complete fusion formula:

$$\mathbf{x}_{k+1|k+1} = \mathbf{P}_{k+1|k+1}((\mathbf{S}\mathbf{P}_{k|k}\mathbf{S}^T + \mathbf{Q})^+ \mathbf{x}_{k+1|k} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{z}_{k+1} + \mathbf{C}_2^+ \mathbf{x}_2) \quad (13)$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{U}_p [\mathbf{U}_p^T ((\mathbf{S}\mathbf{P}_{k|k}\mathbf{S}^T + \mathbf{Q})^+ + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} + \mathbf{C}_2^+) \mathbf{U}_p]^{-1} \mathbf{U}_p^T \quad (14)$$

Observe the symmetry of the solution which *combines all the available knowledge in the information space*. These equations provide a unified fusion of the system dynamics, a subspace model, and measurement noise information. They represent the complete representation of various uncertainties that affect the tracking system.

Compared to a PCA *shape space* representation [7,8], the above formulation uses not only the model subspace (the eigenvectors), but also the actual model distribution, in a unified fusion framework. On the other hand, if only a subspace constraint is desired, we can simply apply the special case of Eq. (2) on Eq. (11), and the resulting *non-orthogonal projection* is still within the same analytical framework.

3 Updating Shape Model: Fusion versus Model Adaptation

The use of a model learned from a pool of training samples to guide a specific case is inherently problematic, especially when novel variations commonly appear. Theoretically, what we really need is *the deformation model of the current case*. Therefore, there is a strong need to update the generic model to reflect what is already known for the current case. A natural choice is to use the initial contour (by hand or through automatic detection) to update the existing model. In the context of the preceding sections, an intriguing question would be *why don't we use fusion on the model and the new contour* by assigning some covariance $\mathbf{C} = \alpha \mathbf{I}$ for the new contour? The answer turns out to be negative and we will get to the reasons at the end of this section.

An alternative tool is incremental PCA (IPCA) [18], but this strategy does not adapt in a sufficiently strong manner. Therefore, we put more emphasis on the new data, and apply a *strongly-adapted-PCA* (SA-PCA) model as follows: We assume that the existing PCA model and the initial contour from the current new case *jointly* represent

the variations of the current case, but with relative energy, or representative power, being α and $(1 - \alpha)$, respectively, with $0 < \alpha < 1$.

If the original covariance matrix \mathbf{C} were stored (when the original dimensionality is not forbiddingly high), the adapted mean \mathbf{x}_m^{new} and covariance matrix \mathbf{C}^{new} would simply be the weighted sum of the two contributing sources:

$$\mathbf{x}_m^{new} = \alpha \mathbf{x}_m + (1 - \alpha) \mathbf{x} \quad (15)$$

$$\begin{aligned} \mathbf{C}^{new} &= \alpha (\mathbf{C} + (\mathbf{x}_m - \mathbf{x}_m^{new})(\mathbf{x}_m - \mathbf{x}_m^{new})^T) + (1 - \alpha)(\mathbf{x} - \mathbf{x}_m^{new})(\mathbf{x} - \mathbf{x}_m^{new})^T \\ &= \alpha \mathbf{C} + \alpha(1 - \alpha)(\mathbf{x} - \mathbf{x}_m)(\mathbf{x} - \mathbf{x}_m)^T \end{aligned} \quad (16)$$

Eigenanalysis can be performed on \mathbf{C}^{new} to obtain the new subspace model.

A more interesting and practical scenario is when \mathbf{C} is not stored and $\{\mathbf{x}_m, \Lambda, \mathbf{U}\}$ resides only *in the subspace*. Denote the subspace component of \mathbf{x} as $\mathbf{x}_s = \mathbf{U}^T \mathbf{x}_d$, where $\mathbf{x}_d = \mathbf{x} - \mathbf{x}_m$, and the residual vector as $\mathbf{x}_r = (\mathbf{x} - \mathbf{x}_m) - \mathbf{U} \mathbf{x}_s$. Let \mathbf{x}_{ru} be the normalized unit vector of \mathbf{x}_r . Through straight algebraic manipulations we can arrive at the adapted eigenanalysis results $\{\mathbf{x}_m^{new}, \Lambda^{new}, \mathbf{U}^{new}\}$ with $\mathbf{U}^{new} = [\mathbf{U}, \mathbf{x}_{ru}] \mathbf{R}$, where \mathbf{R} and Λ^{new} are solutions to the following eigenanalysis problem:

$$\left(\alpha \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \alpha(1 - \alpha) \begin{bmatrix} \mathbf{x}_s \mathbf{x}_s^T & e_r \mathbf{x}_s \\ e_r \mathbf{x}_s^T & e_r^2 \end{bmatrix} \right) \mathbf{R} = \mathbf{R} \Lambda^{new} \quad (17)$$

where $e_r = \mathbf{x}_{ru}^T (\mathbf{x} - \mathbf{x}_m)$ is the residual energy.

The above formulas are extensions of IPCA or eigenspace merging formula of [18], with tunable energy ratios between the new data and the old data. With α set at a smaller value (we use 0.5), the PCA model is strongly adapted toward the current case, hence the name. Now we are ready to point out the differences between fusion and IPCA or SA-PCPA. First of all, a fused model cannot break out of the subspace, while IPCA or SA-PCPA can. More fundamentally, fusion provides the “intersection” of the information sources [19], while IPCA or SA-PCPA yield some “union” of the sources. We need to *augment* instead of *constrain* the generic model, so fusion is not the proper choice.

With SA-PCPA, our framework now incorporates four information sources: the system dynamic, measurement, subspace model, and the initial contour. This last addition is especially useful for periodic shape deformations such as cardiac motion.

4 Implementation, Evaluation, and Analysis

In this paper we test the proposed framework using ultrasound heart sequences. Ultrasound is the noisiest among common medical imaging modalities such as MRI or CT. Echocardiogram (ultrasound heart images) is even worse due to the fast motion of the heart muscle and respiratory interferences [6]. With spatially varying noise characteristics, echocardiograms are ideal for testing our heteroscedastic fusion framework.

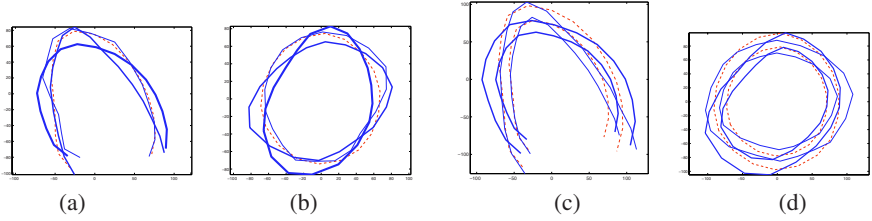


Fig. 2. The dominant eigenshapes for: (a,b) single contour; (c,d) coupled contours; (a,c) Apical views; (b,d) Short axis views. The dashed curves are the model mean.

4.1 Tracking in Echocardiography

We use manually traced left ventricle borders in echocardiography images as the training set. Both apical two- or four-chamber views and parasternal long and short axis views are trained and tested. Landmark points are assigned based on anatomic features (apex, papillary muscles, and septum, etc.). The algorithm can tolerate some variability on the location of the landmark points, partly due to the application of SA-PCA.

The training contours are aligned using the iterative Procrustes analysis approach described by Cootes and Taylor [20] to cancel out global translation, rotation and scaling. PCA is then performed and the original dimensionality is reduced to retain 80-97% of energy, separately tuned for each model. Figure 2 shows the dominant eigenshapes (without splining) for two views along with their model means trained on about 200 contours each for both single and double-contours. A double-contour is treated as a single point in a high-dimensional space.

During testing we assume manual initialization on the first frame, and use a simple dynamic model to impose a temporal smoothness constraint. Without prior knowledge, we employ a diagonal matrix to model the uncertainty in system dynamics, and set the relative confidence of this model empirically. Since we perform alignment on the training shapes before the PCA, at each tracking step the model is aligned to the fusion result $\{\tilde{\mathbf{x}}, \tilde{\mathbf{C}}\}$ using the measurement and the dynamic model. We adopt the optimal transformation \mathcal{T}_o which minimizes a weighted sum-of-squares measure of point difference subject to translation, rotation and scaling ([20], p. 102), with the weighting matrix being $\tilde{\mathbf{C}}^{-1}$. The system transforms the model mean as well as the model covariance using \mathcal{T}_o before the final fusion.

4.2 Motion Estimation with Uncertainty

To measure the motion of each of the control points we use an adaptation of the frame-to-frame motion estimation algorithm described in [13], which has been shown to be very competitive in terms of performance evaluation using standard sequences. We present in the sequel a summary of the algorithm. For more details, see [13].

The main idea is that the *motion in a certain neighborhood can be robustly estimated as the most significant mode of some initial motion estimates* (expressed by mean vectors and associated covariance matrices). The most significant mode is defined by mode



Fig. 3. The 95% confidence ellipses corresponding to the local measurement uncertainty on each control point. The first image shows a single contour for endocardium. The other two show coupled double contours for both endocardium and epicardium.

tracking across scales, while the underlying mechanism for mode detection relies on the variable-bandwidth mean shift [21].

In the current work, for each control point we compute initial estimates using 17×17 windows and fuse the results on $n = 5 \times 5$ neighborhoods. A pyramid of three levels is employed with covariance propagation across levels. Figure 3 depicts the uncertainty calculated at the bottom of the pyramid for the contour points.

To avoid error accumulation from frame to frame, the motion is always computed with reference to the neighborhood of the control point in the first frame of the sequence (i.e., the current frame is always compared to a model extracted from the first frame). Since we update the location of the model at each frame, the motion estimation process always starts with a good initialization, hence the error accumulation is canceled. The overall procedure is suitable for the tracking of periodic sequences such as the heart ultrasound data. It resembles to a template-based tracker, which benefits from the fast computation of frame-to-frame motion.

4.3 Performance Evaluation and Analysis

For systematic evaluation, a set of 30 echocardiogram sequences are used for testing, including parasternal long- and short-axis views and apical two- or four-chamber (AC) views, all with expert-annotated ground-truth contours.

We use two distance measures: the Mean Sum of Squared Distance (MSSD) [22] and the Mean Absolute Distance (MAD) [23]. With consistent results, we report MSSD only in this paper. For the sequence S_i with m frames, $\{c_1, \dots, c_m\}$, where each contour c_j has n points $\{(x_{j,1}, y_{j,1}), \dots, (x_{j,n}, y_{j,n})\}$, the distances to the ground truth S_i^0 are

$$MSSD_i = \frac{1}{m} \sum_{j=1}^m MSSD_{i,j} = \frac{1}{m} \sum_{j=1}^m \frac{1}{n} \sum_{k=1}^n ((x_{j,k} - x_{j,k}^0)^2 + (y_{j,k} - y_{j,k}^0)^2) \quad (18)$$

The overall performance measure for a particular method is the averaged distance on the whole test set of l sequences. A critical difference between our distance measures and those of [22] or [23] is that we have the point correspondence through tracking. As a

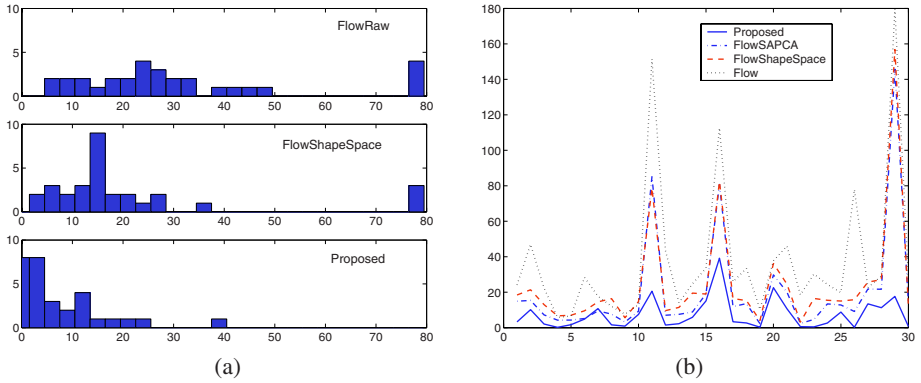


Fig. 4. (a) MSSD histograms over the test set; (b) MSSD curves for the 30 test sequences.

result, we could capture tangent motion components along the contour which can reveal crucial information about cardiac function.

Our proposed framework is compared to three alternatives. The first is a tracker based on the optical flow algorithm without shape constraint (“FlowRaw”) [13]. The second approach is the same tracker but adding orthogonal PCA shape space constraints [8,20,7] (“FlowShapeSpace”). The third is “FlowShapeSpace” but using our SA-PCA model (“FlowSAPCA”) Figure 4 shows the comparison of these methods. Our proposed method (“Proposed”) significantly outperforms others, with an average MSSD of 7.4 ($\sigma = 12.3$) as opposed to 24.3 ($\sigma = 35.4$) by the current approach (“FlowShapeSpace”). Our SA-PCA model alone (“FlowSAPCA”) already brought significant improvement, achieving an average MSSD of 20.4 ($\sigma = 32.8$). The fusion alone (without SA-PCA) had an average MSSD of 18.4 ($\sigma = 22.7$). The MSSD of “FlowRaw” is 38.2 ($\sigma = 83.7$). The combined use of fusion and SA-PCA (i.e., “Proposed”) has apparently brought out a significant performance boost over each alone. Figure 5 shows some tracked sequences. Please also refer to the supplementary videos.

When the measurement process makes a large error in a drop-out or high-noise region, the corresponding localization uncertainty is usually high as well, due to the lack of trackable patterns. Our fusion can correct such errors to a larger extent than what an orthogonal projection can do. This is illustrated by an example in Figure 6.

Our SA-PCA model is especially helpful for shapes that differ significantly from the training set. Figure 7 shows a comparison of IPCA and SA-PCA. In this example, we deliberately used a “wrong” model, i.e., we use the model for apical four chamber (A4C) views (see Figure 2a) to constrain the tracking of this parasternal long axis (PLA) view. PLA views have distinctive patterns that are not seen in apical views (e.g., the upper concave portion). The incremental PCA model, taking in the initial contour (Figure 7a) but with a very small weight ($< 0.01\%$), fails to follow such distinctive patterns; and has constrained the contours to a typical A4C shape (Figure 7b). SA-PCA yields a contour that fits much better to the true border (Figure 7c).

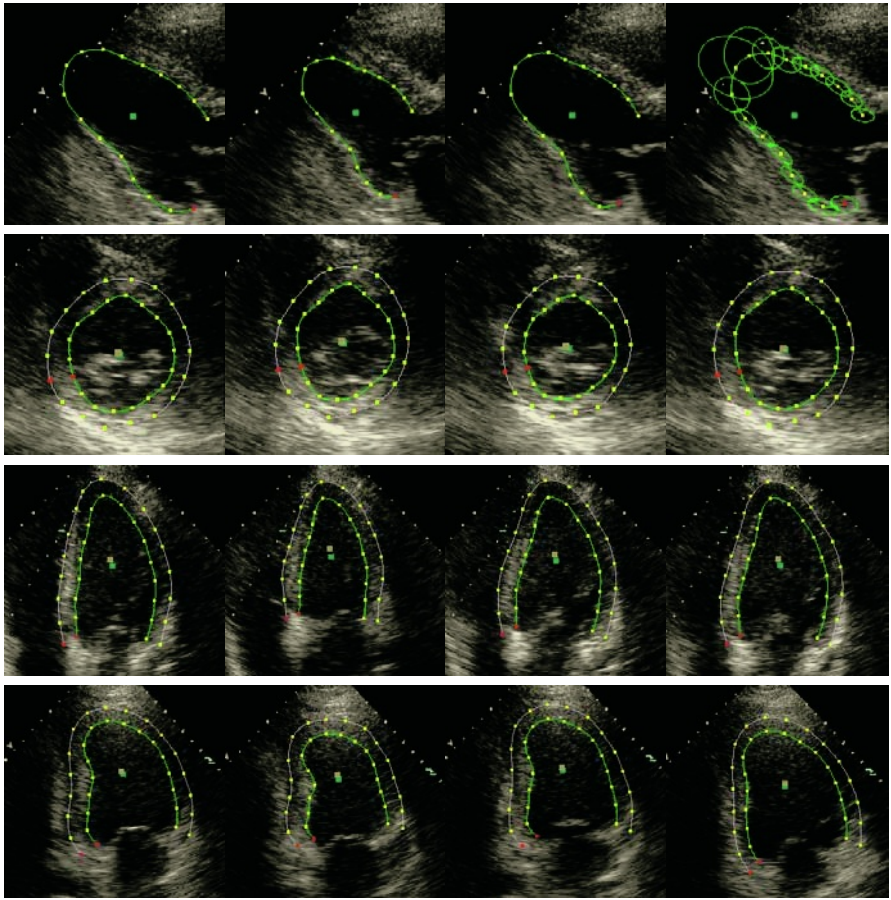


Fig. 5. Four tracking examples in rows, with 4 snapshots per sequence. The frame numbers are 1, 16, 24, and 32; 1, 12, 67, and 81; 1, 12, 18 and 23; 1, 15, 23, and 30; respectively.

4.4 Double-Contour versus Single Contour

Although harder to train a model (requiring more data), coupled double-contours have some advantages: A double-contour approach integrates more spatial information, thus can provide more robust tracking of the two borders. In many cases epicardium is less visible than endocardium (except for the case of pericardial effusion for which the opposite is true!), a double-contour can propagate information from the endocardium to guide the localization of the epicardium (or vice versa). Furthermore, a double-contour can better preserve topology and reduce the chance of crossing (assuming no crossing in the training set). With our explicit constraint from the model distribution using Eq. (9), we limit not only the mode but also the range of shape deformations. Figure 8 shows an example where the double-contour approach clearly improves the performances by single contours alone. Notice the complete appearance change on the right, along with

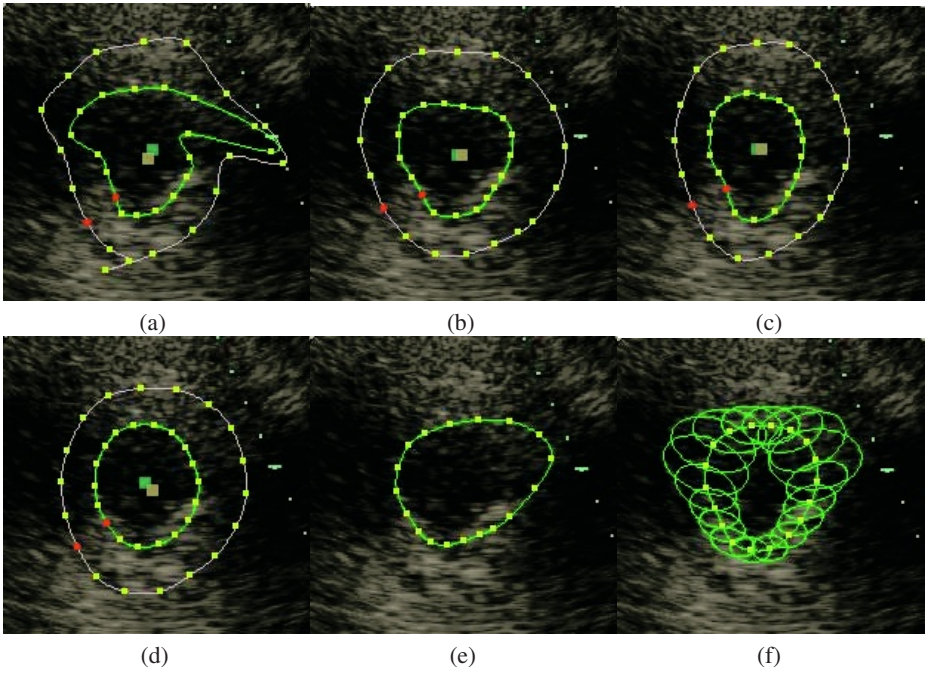


Fig. 6. Comparison of orthogonal projection with our fusion approach in handling large measurement errors. On the same frame we show: (a) the un-constrained flow results; (b) orthogonal projection into the SA-PCA space; (c) result from our fusion framework; (d) by an expert; (e) same as (b) but only for endocardial border; (f) same as (c) for endocardial border (also shown are the uncertainty ellipses). Notice the stronger correction the fusion method brought over the local measurement errors for both single and double contours.

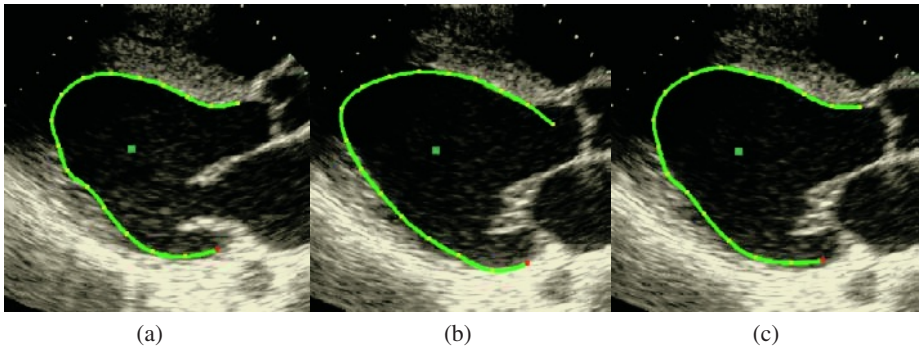


Fig. 7. SA-PCA versus incremental PCA. (a) the initial contour; (b) the 14th frame using an incremental PCA model [18]; (c) the same frame using an SA-PCA model ($\alpha = 0.5$).

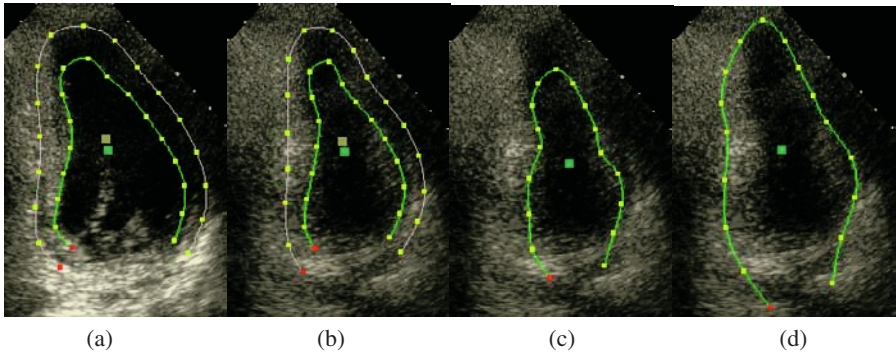


Fig. 8. Double versus single contours. (a) the initial contours; (b) 7th frame tracked by the double-contour; (c) 7th frame with the inner contour; (d) 7th frame with the outer contour.

the large intensity shift at the base, which is in large part to blame for the errors made by the single contours. The double-contour combines information from all locations plus a double-contour model to make the decisions for both contours jointly, thus achieving more robust performance. Our tracker runs at about 30 fps on a 3GHz PC.

5 Related Work

With heteroscedastic measurement noise, an orthogonal projection into the model subspace is not only unjustified, but also damaging in terms of information loss [24]. It can only be justified when the noise is isotropic and homogeneous (*cf.* [20,7,8,2]).

Measurement uncertainty has been exploited for tracking and motion estimation in different contexts. However, none has put all relevant information sources into a unified fusion formulation. Both Brand [24] and Irani [25] use measurement uncertainties, but they did not provide a complete fusion-based tracking framework that combines all the information sources. A rank-constrained flow estimation formulation was proposed by Bregler et al. [26]. They use constraints from both rigid and non-rigid motion represented by basis-shapes. Although occlusion is addressed, measurement uncertainty in general is not optimally exploited. Leedan, Matei, and Meer (e.g. [3]) applied heteroscedastic regression for fitting ellipses and fundamental matrices. The fitting is achieved in the original space with parameterized models. In our formulation, we avoid the parameterization of shape variations. Instead, we build subspace probabilistic models through PCA and obtain closed-form solutions on both the mean and covariance of the fitted data. Although simple, this model proves to be flexible and powerful for the current application, especially with the use of an SA-PCA model. Nevertheless, for applications where this model is too restrictive, a future research is to integrate more sophisticated nonlinear models. (e.g., [27]). Robust model matching [28] relying on M-estimators or RANSAC has been applied to limit or eliminate the influence of data components that are outliers with respect to the model. Again, the locally (in space or time) varying uncertainties are not exploited in these frameworks.

There is much research work done in medical domain that tracks heart motion using various techniques (e.g., [8,23,22], etc.). However, none of these addresses the issue of heteroscedastic noise and its fusion with other information sources.

6 Conclusions and Future Work

This paper presented a joint information fusion framework to track shapes under heteroscedastic noise with a strongly adapted subspace model constraint.

Extensions to 3D or 2D+T(time) are natural. Extension to triple(or more)-contours are feasible if sufficient training data are available. Our framework is general and can be applied to other applications. Considering the heavy noise situation in the ultrasound data, success of this approach on other data is expected in our future efforts.

Acknowledgment. We would like to thank Visvanathan Ramesh from Siemens Corporate Research for insightful discussions on the subject. We are grateful to Alok Gupta from the CAD Group, Siemens Medical Solutions for his support and encouragement.

References

1. Cootes, T., Taylor, C.: Active shape models-'smart snakes'. In: Proc. British Machine Vision Conference. (1992) 266–275
2. Turk, M.A., Pentland, A.P.: Face recognition using eigen-face. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Hawaii. (1991) 586–591
3. Leedan, Y., Meer, P.: Heteroscedastic regression in computer vision: Problems with bilinear constraint. *Intl. J. of Computer Vision* **37** (2000) 127–150
4. Kanazawa, Y., Kanatani, K.: Do we really have to consider covariance matrices for image features? In: Proc. Intl. Conf. on Computer Vision, Vancouver, Canada. Volume II. (2001) 586–591
5. Irani, M., Anandan, P.: Factorization with uncertainty. In: Proc. 6th European Conf. on Computer Vision, Dublin, Ireland. (2000) 539–553
6. Oh, J.K., Seward, J.B., Tajik, A.J.: *The Echo Manual*. Lippincott Williams & Wilkins, Philadelphia, (1999)
7. Blake, A., Isard, M.: *Active contours*. Springer Verlag (1998)
8. Jacob, G., Noble, A., Blake, A.: Robust contour tracking in echocardiographic sequence. In: Proc. Intl. Conf. on Computer Vision, Bombay, India. (1998) 408–413
9. Paragios, N.: A variational approach for the segmentation of the left ventricle in cardiac images. In: Proc. IEEE Workshop on Variational and Level Set Methods in Computer Vision, Vancouver, Canada. (2001)
10. Goldenberg, R., Kimmel, R., Rivlin, E., Rudzsky, M.: Cortex segmentation: A fast variational geometric approach. *IEEE Trans. Medical Imaging* **21** (2002) 1544–51
11. Wang, S., Ji, X., Liang, Z.P.: Landmark-based shape deformation with topology-preserving constraints. In: Proc. Intl. Conf. on Computer Vision, Nice, France. (2003)
12. Black, M., Jepson, A.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In: Proc. European Conf. on Computer Vision, Cambridge, UK. (1996) 610–619

13. Comaniciu, D.: Nonparametric information fusion for motion estimation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Madison, Wisconsin. Volume I. (2003) 59–66
14. Scharf, L.L.: Statistical Signal Processing. Addison Wesley, Reading, MA (1991)
15. Bar-Shalom, Y., Campo, L.: The effect of the common process noise on the two-sensor fused track covariance. IEEE Trans. Aero. Elect. Syst. **AES-22** (1986) 803–805
16. Li, X., Zhu, Y., Han, C.: Unified optimal linear estimation fusion - part i: Unified models and fusion rules. In: Proc. of 3rd Intl. Conf. on Information Fusion, Paris, France. (2000) MoC2–10–MoC2–17
17. Anderson, B., Moore, J.: Optimal filtering. Prentice-Hall (1979)
18. Hall, P., Marshall, D., Martin, R.: Merging and splitting eigenspace models. IEEE Trans. Pattern Anal. Machine Intell. **22** (2000) 1042–1048
19. Julier, S., Uhlmann, J.: A non-divergent estimation algorithm in the presence of unknown correlations. In: Proc. American Control Conf., Albuquerque, NM. (1997)
20. Cootes, T., Taylor, C.: Statistical models for appearance for computer vision. (2001) Unpublished manuscript. Available at http://www.wiau.man.ac.uk/~bim/Models/app_model.ps.gz.
21. Comaniciu, D.: An algorithm for data-driven bandwidth selection. IEEE Trans. Pattern Anal. Machine Intell. **25** (2003) 281–288
22. Akgul, Y., Kambhamettu, C.: A coarse-to-fine deformable contour optimization framework. IEEE Trans. Pattern Anal. Machine Intell. **25** (2003) 174–186
23. Mikić, I., Krucinski, S., Thomas, J.D.: Segmentation and tracking in echocardiographic sequences: Active contours guided by optical flow estimates. IEEE Trans. Medical Imaging **17** (1998) 274–284
24. Brand, M., Bhotika, R.: Flexible flow for 3D nonrigid object tracking and shape recovery. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Hawaii. Volume I. (2001) 315–322
25. Irani, M.: Multi-frame optical flow estimation using subspace constraints. In: Proc. Intl. Conf. on Computer Vision, Kerkyra, Greece. (1999) 626–633
26. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3d shape from image streams. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head, SC. Volume II. (2000) 690–696
27. Cremers, D., Kohlberger, T., Schnorr, C.: Nonlinear shape statistics in mumford-shah based segmentation. In: Proc. European Conf. on Computer Vision, Copenhagen, Denmark. (2002) 93–108
28. Black, M., Ananadan, P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. Computer Vision and Image Understanding **63** (1996) 75–104

A Statistical Model for General Contextual Object Recognition

Peter Carbonetto¹, Nando de Freitas¹, and Kobus Barnard²

¹ Dept. of Computer Science, University of British Columbia
Vancouver, Canada
{pcarbo, nando}@cs.ubc.ca

² Dept. of Computer Science, University of Arizona
Tucson, Arizona
kobus@cs.arizona.edu

Abstract. We consider object recognition as the process of attaching meaningful labels to specific regions of an image, and propose a model that learns spatial relationships between objects. Given a set of images and their associated text (e.g. keywords, captions, descriptions), the objective is to segment an image, in either a crude or sophisticated fashion, then to find the proper associations between words and regions. Previous models are limited by the scope of the representation. In particular, they fail to exploit spatial context in the images and words. We develop a more expressive model that takes this into account. We formulate a spatially consistent probabilistic mapping between continuous image feature vectors and the supplied word tokens. By learning both word-to-region associations and object relations, the proposed model augments scene segmentations due to smoothing implicit in spatial consistency. Context introduces cycles to the undirected graph, so we cannot rely on a straightforward implementation of the EM algorithm for estimating the model parameters and densities of the unknown alignment variables. Instead, we develop an approximate EM algorithm that uses loopy belief propagation in the inference step and iterative scaling on the pseudo-likelihood approximation in the parameter update step. The experiments indicate that our approximate inference and learning algorithm converges to good local solutions. Experiments on a diverse array of images show that spatial context considerably improves the accuracy of object recognition. Most significantly, spatial context combined with a nonlinear discrete object representation allows our models to cope well with over-segmented scenes.

1 Introduction

The computer vision community has invested a great deal of effort toward the problem of recognising objects, especially in recent years. However, less attention has been paid to formulating an understanding of general object recognition; that is, properly isolating and identifying classes of objects (e.g. ceiling, polar bear) in an agent's environment. We say an object is *recognised* when it is labeled with a concept in an appropriate and consistent fashion. This allows us



Fig. 1. Examples of images paired with their captions. A crucial point is that the model has to learn which word belongs with which part of the image.

to propose a practical answer to the question of what is an object: an object is a semantic concept (in our case, a noun) in an image caption. Pursuing general object recognition may appear to be premature, given that good unconstrained object representations remain elusive. However, we maintain that a principled exploration using simple, learned representations can offer insight for further direction. Our approach permits examination of the relations between high-level computer vision and language understanding.

Ideally, we would train our system on images where the objects have been properly segmented and accurately labeled. However, the collection of supervised data by manually labeling semantically-contiguous regions of images is both time-consuming and problematic. We require captions at an image level, not at an image region level, and as a result we have large quantities of data at our disposal (e.g. thousands of Corel images with keywords, museum images with meta data, news photos with captions, and Internet photo stock agencies). Previous work shows that it is reasonable to use such loosely labeled data for problems in vision and image retrieval[1,4,13,11,2,7]. We stress that throughout this paper we use annotations solely for testing — training data includes *only* the text associated with entire images. We do so at a cost since we are no longer blessed with the exact associations between objects and semantic concepts. In order to learn a model that *annotates*, *labels* or *classifies* objects in a scene, training implicates finding the *associations*, *alignments* or *correspondence* between objects and concepts in the data. As a result, the learning problem is unsupervised (or semi-supervised). We adapt the work in another unsupervised problem — learning a lexicon from an aligned bitext in statistical machine translation [9] — to general object recognition, as first proposed in [13].

The data consists of images paired with associated text. Each image consists of a set of blobs that identify the objects in the scene. A *blob* is a set of features that describes an object. Note that this does not imply that the scene is necessarily segmented, and one could easily implement scale-invariant descriptors to represent object classes, as in [14,12]. Abstractly, a caption consists of a bag of semantic concepts that describes the objects contained in the image scene. For the time being, we restrict the set of concepts to English nouns (e.g. “bear”, “floor”). See Fig. 1 for some examples of images paired with their captions.

We make three major contributions in this paper.

Our first contribution is to address a limitation of existing approaches for translating image regions to words: the assumption that blobs are statistically

independent, usually made to simplify computation. Our model relaxes this assumption and allows for interactions between blobs through a Markov random field (MRF). That is, the probability of an image blob being aligned to a particular word depends on the word assignments of its neighbouring blobs. Due to the Markov assumption, we still retain some structure. One could further introduce relations at different scales using a hierarchical representation, as in [15].

Dependence between neighbouring objects introduces spatial context to the classification. Spatial context increases expressiveness; two words may be indistinguishable using low-level features such as colour (e.g. “sky” and “water”) but neighbouring objects may help resolve the classification (e.g. “airplane”). Context also alleviates some of the problems caused by a poor blob clustering. For example, birds tend to be segmented into many parts, which inevitably get placed in separate bins due to their different colours. The contextual model can learn the co-occurrence of these blobs and increase the probability of classifying them as “bird” when they appear next to each other in a scene. Experiments in Sect. 4 confirm our intuition, that spatial context combined with a basic nonlinear decision boundary produces relatively accurate object annotations.

Second, we propose an approximate algorithm for estimating the parameters when the model is not completely observed and the partition function is intractable. Like previous work on detection of man-made structures using MRFs [16,17], we use pseudo-likelihood for parameter estimation, although we go further and consider the unsupervised setting in which we learn both the potentials and the labels. As with most algorithms based on loopy belief propagation, our algorithm has no theoretical guarantees of convergence, but empirical trials show reasonably stable convergence to local solutions.

Third, we discuss how the contextual model offers purchase on the image segmentation problem. Segmentation algorithms commonly over-segment because low-level features are insufficient for forming accurate boundaries between objects. The object recognition data has semantic information in the form of captions, so it is reasonable to expect that additional high-level information could improve segmentations. Barnard *et al.* [3] show that translation models can suggest appropriate blob merges based on word predictions. For instance, high-level groupings can link the black and white halves of a penguin. Spatial consistency learned with semantic information smooths labellings, and therefore our proposed contextual model learns to cope with over-segmented images. In fact, with this model, a plausible strategy is to start with image grid patches and let segmentations emerge as part of the labeling process (see Fig. 6).

2 Specification of Contextual Model

First, we introduce some notation. The observed variables are the words w_{n1}, \dots, w_{nL_n} and the blobs b_{n1}, \dots, b_{nM_n} paired with each image (or document) n . M_n is the number of blobs or regions in image n , and L_n is the size of the image caption. For each blob b_{nu} in image n , we need to align it to a word from the attached caption. The unknown association is represented by the variable

a_{nu} , such that $a_{nu} = i$ if and only if blob b_{nu} corresponds to word w_{ni} . The sets of words, blobs and alignments for all documents are denoted by w , b and a , respectively. Each w_{ni} represents a separate concept or object from the set $\{1, \dots, W\}$, where W is the total number of word tokens.

Results in [11] suggest that representation using a mixture of Gaussians facilitates the data association task and improves object recognition performance. However, we retain the blob discretisation proposed by [13] because it scales better to large data sets and we will find model computation easier to manage. We use k -means to assign each blob b_{nu} in the feature space \mathbb{R}^F to one of the B clusters. F is the number of features and B is the number of blob tokens.

The translation lexicon is a $B \times W$ table with entries $t(b^*|w^*)$, where w^* denotes a particular word token and b^* denotes a particular blob token. We define ψ to be a $W \times W$ table of potentials describing the “next to” relation between blob annotations. We define spatial context to be symmetric, so $\psi(w^*, w^\diamond) = \psi(w^\diamond, w^*)$. The set of model parameters is $\theta \triangleq \{t, \psi\}$. The set of cliques in document n is denoted by C_n . The complete likelihood over all the documents is

$$p(b, a | w, \theta) = \prod_{n=1}^N \frac{1}{Z_n(\theta)} \prod_{u=1}^{M_n} \Phi(b_{nu}, a_{nu}) \prod_{(u,v) \in C_n} \Psi(a_{nu}, a_{nv}) \quad (1)$$

where we define the potentials over the translation and spatial context cliques to be

$$\begin{aligned} \Phi(b_{nu}, a_{nu}) &= \prod_{i=1}^{L_n} t(b_{nu}, w_{ni})^{\delta(a_{nu}=i)} \\ \Psi(a_{nu}, a_{nv}) &= \prod_{i=1}^{L_n} \prod_{j=1}^{L_n} \psi(w_{ni}, w_{nj})^{\delta(a_{nu}=i) \times \delta(a_{nv}=j)} . \end{aligned}$$

$Z_n(\theta)$ is the partition function for the disjoint graph of document n . δ is the indicator function such that $\delta(a_{nu} = i)$ is 1 if and only if $a_{nu} = i$, and 0 otherwise. Fig. 2 shows the undirected graphical model for a single example document with six blobs.

3 Model Computation

Spatial context improves expressiveness, but this comes at an elevated computational cost due to cycles introduced in the undirected graph. We use a variation of Expectation Maximisation (EM) for computing an approximate maximum likelihood estimate. In the E Step, we use loopy belief propagation [19] on the complete likelihood (1) to compute the marginals $\tilde{p}(a_{nu} = i)$ and $\tilde{p}(a_{nu} = i, a_{nv} = j)$. Since the partition function is intractable and the potentials over the cliques are not complete, parameter estimation in the M Step is difficult. Iterative scaling (IS) works on arbitrary exponential models, but it is not a saving grace because convergence is exponentially-slow. An alternative to

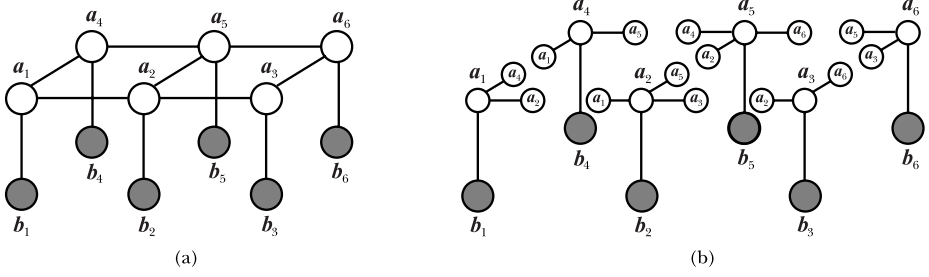


Fig. 2. (a) A sample Markov random field for one document with 6 blob sites. We have omitted the n subscript. The Φ potentials are defined on the vertical lines, and the Ψ potentials are defined on the horizontal lines. (b) The corresponding pseudo-likelihood approximation.

the maximum likelihood estimator is the pseudo-likelihood [6], which maximises local neighbourhood conditional probabilities at sites in the MRF, independent of other sites. The conditionals over the neighbourhoods of the vertices allow the partition function to decouple and render parameter estimation tractable. The pseudo-likelihood neglects long-range interactions, but empirical trials show reasonable and consistent results [20].

Essentially, the pseudo-likelihood is a product of undirected models, where each undirected model is a single latent variable a_{nu} and its observed partner b_{nu} conditioned on the variables in its Markov blanket. See Fig. 2 for an example. The pseudo-likelihood approximation of (1) is

$$p\ell(b, a \mid w, \theta) = \prod_{n=1}^N \prod_{u=1}^{M_n} \frac{1}{Z_{nu}(\theta)} \Phi(b_{nu}, a_{nu}) \prod_{v \in \mathcal{N}_{nu}} \Psi(a_{nu}, a_{nv}) \quad (2)$$

where \mathcal{N}_{nu} is the set of blobs adjacent to node u and $Z_{nu}(\theta)$ is the partition function for the neighbourhood at site u in document n .

Iterative scaling allows for a tractable update step by bounding the log pseudo-likelihood. As in [5], we take the partial derivative of a tractable lower bound, $\Lambda(\theta)$, with respect to the model parameters, resulting in the equations

$$\begin{aligned} \frac{\partial \Lambda}{\partial t(b^*, w^*)} &= \sum_{n=1}^N \sum_{u=1}^{M_n} \sum_{i=1}^{L_n} \delta(b_{nj} = b^*) \delta(w_{ni} = w^*) \tilde{p}(a_{nu} = i) \\ &+ \sum_{n=1}^N \sum_{u=1}^{M_n} \Delta t(b^*, w^*)^{|\mathcal{N}_{nu}|+1} \sum_{i=1}^{L_n} \delta(w_{ni} = w^*) p(b_{nu} = b^*, a_{nu} = i \mid \theta) \end{aligned} \quad (3)$$

$$\begin{aligned} \frac{\partial \Lambda}{\partial \psi(w^*, w^\diamond)} &= \sum_{n=1}^N \sum_{u=1}^{M_n} \sum_{v \in \mathcal{N}_{nu}} \sum_{i=1}^{L_n} \sum_{j=1}^{L_n} \delta(w_{ni} = w^*) \delta(w_{nj} = w^\diamond) \tilde{p}(a_{nu} = i, a_{nv} = j) \\ &+ \sum_{n=1}^N \sum_{u=1}^{M_n} \sum_{v \in \mathcal{N}_{nu}} \Delta \psi(w^*, w^\diamond)^{|\mathcal{N}_{nu}|+1} \sum_{i=1}^{L_n} \sum_{j=1}^{L_n} \delta(w_{ni} = w^*) \delta(w_{nj} = w^\diamond) p(a_{nu} = i \mid \tilde{a}_{nv} = j, \theta) \end{aligned} \quad (4)$$

where we take $p(a_{nu} = i | \tilde{a}_{nv} = j, \theta)$ to be the estimate of alignment $a_{nu} = i$ conditional on the empirical distribution $\tilde{p}(a_{nv} = j)$ and the current parameters. To find the conditionals for (4), we run universal propagation and scaling (UPS) [23] at each pseudo-likelihood site nu with the neighbours $v \in \mathcal{N}_{nu}$ clamped to the current marginals $\tilde{p}(a_{nv})$. UPS is exact because the undirected graph at each neighbourhood is a tree. Also note that (3) requires estimation of the blob densities in addition to the alignment marginals.

The partial derivatives do not decouple because we cannot expect the feature counts (i.e. the number of cliques) to be the same for every site neighbourhood. Observing that (3) and (4) are polynomial expressions where each term has degree $|\mathcal{N}_{nu}| + 1$, we can find new parameter estimates by plugging the solution for (3) or (4) into the IS update $\theta_i^{(new)} = \theta_i \times \Delta \theta_i$. Cadez and Smyth [10] prove that the gradient of the pseudo-likelihood with respect to a global parameter is indeed well-conditioned since it has a unique positive root.

On large data sets, the IS updates are slow. Optionally, one can boost the M Step with an additional iterative proportional fitting (IPF) step, which converges faster than IS because it doesn't have to bound the gradient of the log likelihood. We are permitted to perform an IPF update on t because it is associated with only one clique in each neighbourhood. The IPF update for t is

$$t^{(new)}(b^*, w^*) = t(b^*, w^*) \times \frac{\sum_{n=1}^N \sum_{u=1}^{M_n} \sum_{i=1}^{L_n} \delta(w_{ni} = w^*) \delta(b_{nu} = b^*) \tilde{p}(a_{nu} = i)}{\sum_{n=1}^N \sum_{u=1}^{M_n} \sum_{i=1}^{L_n} \delta(w_{ni} = w^*) p(b_{nj} = b^*, a_{nu} = i | \theta)} . \quad (5)$$

To stabilise the parameter updates, we place weak priors on t and ψ of 10^{-5} and 10^{-4} , respectively. We find a near-uninformative prior for ψ works well, although we caution that prior selection in MRFs is notoriously difficult [6].

4 Experiments

The experiments compare two models. *dInd* is the discrete translation model proposed in [13] and assumes object annotations are independent. *dMRF* is the contextual model developed in this paper. We evaluate object recognition results on data sets composed of a variety of images, and examine the effect of two different segmentations on the performance of our models.

We composed two sets, denoted by *CorelB* and *CorelC*¹. The first data set, *CorelB*, has 199 training images and 100 test images, with 38 words in the training set. The *CorelB* data set contains a total of 504 annotated images, divided into training and test sets numbering 336 and 168 in size. The training set has a total of 55 distinct concepts. The frequencies of words in the *CorelB* labels and manual annotations are shown in Fig. 4.

We consider two scenarios. In the first, we use Normalized Cuts [22] to segment the images. In the second scenario, we take on the object recognition task without the aid of a sophisticated segmentation algorithm, and instead construct a uniform grid of patches over the image. Examples of the segmentations

¹ The experiment data is available at <http://www.cs.ubc.ca/~pcarbo>.

are shown in Fig. 6. The choice of grid size is important since the features are not scale invariant. We use patches approximately 1/6th the size of the image; smaller patches introduce too much noise to the features, and larger patches contain too many objects. The two scenarios are denoted by *NCuts* and *Grid*.

The blobs are described using simple colour features. Vertical position was found to be a simple and useful feature [11], but it does not work well with the discrete models because the K-means clustering tends to be poor. The number of blob clusters, B , is a significant factor; too small, and the classification is non-separable; too large, and finding the correct associations is near impossible. As a rule of thumb, we found $B = 5W$ to work well.

The relative importance of objects in a scene is task-dependent. Ideally, when collecting user-annotated images for evaluation, we should tag each word with a weight to specify its prominence in the scene. In practice, this is problematic because different users focus their attention on different concepts, not to mention the fact that it is a burdensome task. Rewarding prediction accuracy over blobs — not objects — is a reasonable performance metric as it matches the objective functions of the translation models. We have yet to compare our models using the evaluation procedures proposed in [8,2]. The prediction error is given by

$$\frac{1}{N} \sum_{n=1}^N \frac{1}{M_n} \sum_{u=1}^{M_n} \left(1 - \delta \left(\hat{a}_{nu} = a_{nu}^{(max)} \right) \right) \quad (6)$$

where $a_{nu}^{(max)}$ is the model alignment with the highest probability and \hat{a}_{nu} is the ground-truth annotation.

One caveat regarding our evaluation procedure: the segmentation scenarios are not directly comparable because the manual annotation data is slightly different for *NCuts* and *Grid*. For testing purposes, we annotated the image segments in the two scenarios by hand. Since we cannot expect the segmentation methods to perfectly delineate concepts in a scene, a single region may contain several subjects, all deemed to be correct. We found that the Normalized Cuts segments frequently encompassed several objects, whereas the uniform grid segments, by virtue of being smaller, more regularly contained a single object. As a result, our evaluation measure can report the same error for the two scenarios, when in actual fact the uniform grid produces more precise object recognition. To address the role of segmentation in object recognition more faithfully, we are in the process of building data sets with ground-truth annotations and segmentations.

Figure 3 compares model annotations over 12 trials with different initialisations. Model *dInd* took on the order of a few minutes to converge to a local minimum of the log-likelihood, whereas model *dMRF* generally took several hours to learn the potentials. The first striking observation is that the contextual model shows consistently improved results over *dInd*. Additionally, the variance of *dMRF* is not high, despite an increase in the number of parameters and a lack of convergence guarantees.

Recognition using the grid segmentation tends to do better than the Normalized Cuts results, keeping in mind that we require the *Grid* annotations to be more precise, as discussed above. This suggests we can achieve comparable

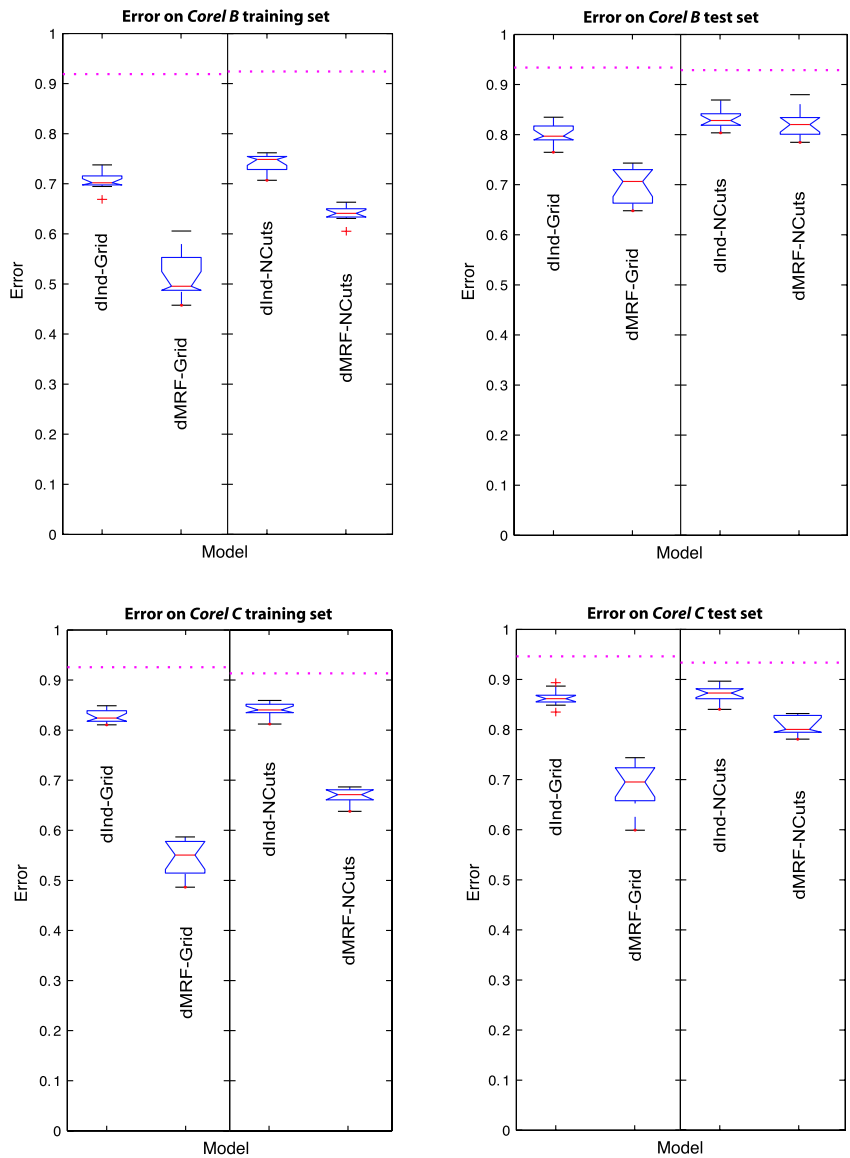


Fig. 3. Prediction error of the two models using the *Grid* and *NCuts* segmentations on the *CorelB* and *CorelC* data sets. The results are displayed using a Box-and-Whisker plot. The middle line of a box is the median. The central box represents the values from the 25 to 75 percentile, using the upper and lower statistical medians. The horizontal line extends from the minimum to the maximum value, excluding outside and far out values which are displayed as separate points. The dotted line at the top is the random upper bound. The contextual model introduced in this paper substantively reduces the error over *dInd* in the grid segmentation case.

Precision on *CorelB* data set using *Grid*

WORD	LABEL %		ANNOTATION %†		<i>dInd</i> PR.		<i>dMRF</i> PR.	
	TRAIN	TEST†	TRAIN	TEST	TRAIN	TEST	TRAIN	TEST
airplane	0.060	0.055	0.036	0.028	0.135	0.102	0.290	0.187
astronaut	0.003	0.003	0.001	0.002	0.794	0.087	0.000	0.135
atm	n/a	0.003	n/a	0.006	n/a	0.000	n/a	0.000
bear	0.031	0.017	0.021	0.013	0.192	0.092	0.452	0.272
beluga	n/a	0.003	n/a	0.005	n/a	0.000	n/a	0.000
bill	0.019	0.017	0.046	0.031	0.269	0.175	0.335	0.146
bird	0.017	0.010	0.009	0.004	0.148	0.111	0.556	0.458
building	0.014	0.007	0.006	0.002	0.368	0.013	0.408	0.137
cheetah	0.012	0.017	0.010	0.013	0.833	0.683	0.710	0.395
cloud	0.050	0.045	0.050	0.048	0.222	0.152	0.300	0.239
coin	0.005	0.007	0.008	0.008	0.611	0.213	0.767	0.017
coral	0.005	n/a	0.011	n/a	0.815	n/a	0.738	n/a
crab	0.002	0.003	0.001	0.002	0.802	0.663	1.000	0.833
dolphin	0.014	0.003	0.006	0.001	0.606	0.899	0.916	0.000
earth	0.003	0.007	0.004	0.002	0.543	0.000	0.732	0.142
fish	0.007	n/a	0.003	n/a	0.236	n/a	0.695	n/a
flag	0.005	0.007	0.004	0.008	0.617	0.831	0.888	0.890
flowers	n/a	0.003	n/a	0.003	n/a	0.000	n/a	0.000
fox	0.010	0.017	0.011	0.010	0.246	0.052	0.691	0.008
goat	0.003	n/a	0.001	n/a	0.704	n/a	0.994	n/a
grass	0.129	0.120	0.177	0.157	0.165	0.176	0.172	0.229
hand	n/a	0.003	n/a	0.002	n/a	0.000	n/a	0.000
map	n/a	0.003	n/a	0.003	n/a	0.000	n/a	0.000
mountain	0.012	0.003	0.007	0.000	0.204	0.060	0.671	0.057
person	0.003	0.014	0.001	0.004	0.170	0.037	1.000	0.000
polarbear	0.021	0.024	0.016	0.015	0.510	0.625	0.681	0.634
rabbit	0.002	n/a	0.001	n/a	0.489	n/a	1.000	n/a
road	0.026	0.024	0.016	0.008	0.190	0.062	0.526	0.213
rock	0.019	0.038	0.018	0.033	0.127	0.078	0.446	0.130
sand	0.034	0.024	0.023	0.024	0.246	0.150	0.330	0.185
shuttle	0.007	0.007	0.006	0.005	0.504	0.268	0.305	0.107
sky	0.156	0.137	0.172	0.173	0.190	0.138	0.190	0.208
snow	0.036	0.062	0.064	0.110	0.358	0.296	0.435	0.356
space	0.007	0.010	0.008	0.018	0.000	0.000	0.326	0.071
tiger	0.021	0.034	0.015	0.030	0.450	0.233	0.623	0.285
tracks	0.024	0.017	0.010	0.009	0.351	0.163	0.575	0.315
train	0.026	0.021	0.021	0.014	0.165	0.164	0.396	0.272
trees	0.095	0.076	0.075	0.069	0.169	0.094	0.227	0.134
trunk	0.003	0.010	0.001	0.006	0.553	0.023	0.910	0.000
water	0.091	0.089	0.120	0.095	0.214	0.133	0.212	0.137
whale	0.007	0.007	0.003	0.004	0.476	0.268	0.854	0.405
wolf	0.009	0.038	0.007	0.029	0.512	0.166	0.660	0.102
zebra	0.012	0.010	0.009	0.007	0.652	0.667	0.903	0.710
Totals	1.000	1.000	1.000	1.000	0.294	0.199	0.486	0.301

Fig. 4. The first four columns list the probability of finding a particular word in an image caption and manually-annotated image region in the *CorelB* data using the grid segmentation. The final four columns show the precision of models *dInd* and *dMRF* averaged over the 12 trials. Precision is defined as the probability the model’s prediction is correct for a particular word and blob. Since precision is 1 minus the error of (6), the total precision on both the training and test sets matches the average performance shown in Fig. 3. The variance in the precision on individual words is not presented in this table. Note that some words do not appear in both the training and test sets, hence the “n/a”.

† We underline the fact that an agent would not have access to the test image labels and information presented in the ANNOTATION % column.

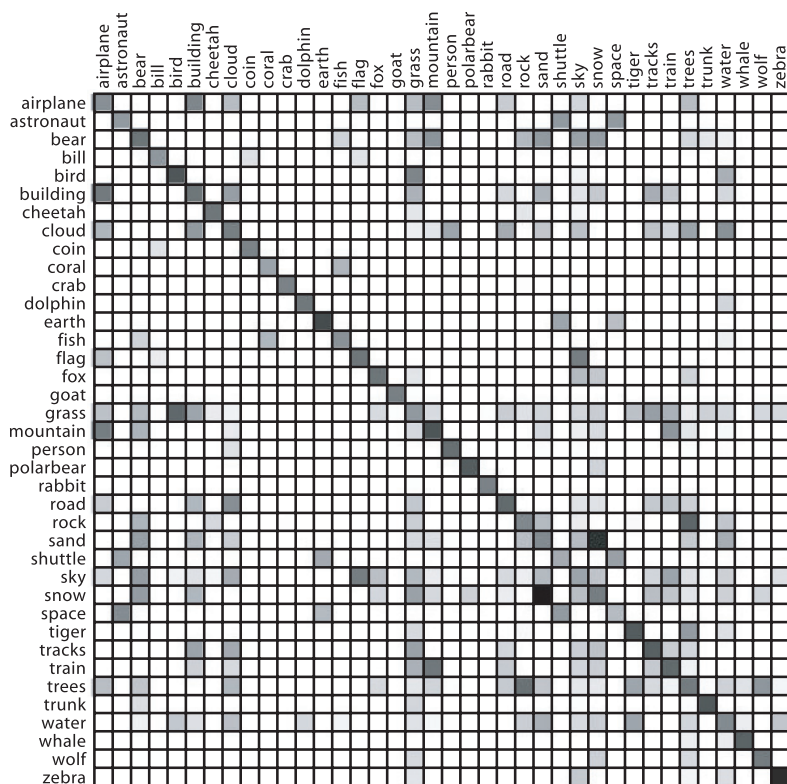


Fig. 5. Our algorithm learned the above contextual relations for the *CorelB* data using the grid segmentation (the matrix is averaged over all the learning trials). Darker squares indicate a strong neighbour relationship between concepts. White indicates that the words were never observed next to each other. For example, fish goes with coral. It is intriguing that planes go with buildings!

results without an expensive segmentation step. However, we are cautious not to make strong claims about the utility of sophisticated low-level segmentations in object recognition because we do not have a uniform evaluation framework nor have we examined segmentation methods in sufficient variety and detail.

Figure 4 shows the precision on individual words for the *CorelB Grid* experiment, averaged over the 12 trials. While not shown in the figures, we have noticed considerable variation among individual trials as to what words are predicted with high precision. For example, model *dMRF* with a grid segmentation predicts the word “train” with average success 0.396, although the precision on individual trials ranges from 0.102 to 0.862 in the training set. Significantly, the spatial context model tends to do better on words that cannot be described using simple colour features, such as “building”, “bear” and “airplane”.

Figure 5 depicts the potentials ψ for the *CorelB Grid* experiment. Note the table is symmetric. The diagonal is dark because words appear most often next

Selected annotations predicted by the *dMRF* model on the *CorelB* training and test sets are displayed in Fig. 6. In several instances, observers found *dInd* annotations more appealing than those of *dMRF*, even though precision using the latter model was higher. This is in part because *dMRF* tends to be more accurate for the background (e.g. sky), whereas observers prefer getting the principal subject (e.g. airplane) correct. This suggests that we should explore alternative evaluation measures based on decision theory and subjective prior knowledge.

5 Discussion and Conclusions

We showed that spatial context helps classify objects, especially when blob features are ineffective. Poorly-classified words may be easier to label when paired with easily-separable concepts. Spatial context purges insecure predictions, and thus acts as a smoothing agent for scene annotations. The pseudo-likelihood approximation allows for satisfactory results, but we cannot precisely gauge the extent to which it skews parameters to suboptimal values. Our intuition is that it gives undue preference to the diagonal elements of the spatial context potentials.

Normalized Cuts is widely considered to produce good segmentations of scenes, and surprisingly our experiments indicate that crude segmentations work equally well or better for object recognition. Upon further consideration, our results are indeed sensible. We are attempting to achieve an optimal balance between loss of information through compression and adeptness in data association through mutual information between blobs and labels. The Normalized Cuts settings we use tend to fuse blobs containing many objects, which introduces noise to the classification data. *dMRF* can cope with lower levels of compression, and hence it performs much better with smaller segments even if they ignore object boundaries. Since model *dMRF* fuses blobs with high affinities, we claim it is a small step towards a model that learns both scene segmentations and annotations concurrently. A couple considerable obstacles in the development of such a model are the design of efficient training algorithms and the creation of evaluation schemes that uniformly evaluate the quality of segmentations combined with annotations.

Acknowledgements. We thank Yee Whye Teh for his discussions on parameter estimation in graphical models and Kevin Murphy for his advice on belief propagation. We would also like to acknowledge invaluable financial support from IRIS ROPAR and NSERC.

References

1. Barnard, K., Duygulu, P., Forsyth, D.A.: Clustering art. IEEE Conf. Comp. Vision and Pattern Recognition (2001)
2. Barnard, K., Duygulu, P., Forsyth, D.A., de Freitas, N., Blei, D.M., Jordan, M.I.: Matching words and pictures. J. Machine Learning Res., Vol. 3 (2003) 1107-1135

3. Barnard, K., Duygulu, P., Guru, R., Gabbur, P., Forsyth, D.A.: The Effects of segmentation and feature choice in a translation model of object recognition. *IEEE Conf. Comp. Vision and Pattern Recognition* (2003)
4. Barnard, K., Forsyth, D.A.: Learning the semantics of words and pictures. *Intl. Conf. Comp. Vision* (2001)
5. Berger, A.: The Improved iterative scaling algorithm: a gentle introduction. Carnegie Mellon University (1997)
6. Besag, J.: On the Statistical analysis of dirty pictures. *J. Royal Statistical Society, Series B*, Vol. 48, No. 3 (1986) 259–302
7. Blei, D.M., Jordan, M.I.: Modeling annotated data. *ACM SIGIR Conf. on Research and Development in Information Retrieval* (2003)
8. Borra, S., Sarkar, S.: A Framework for performance characterization of intermediate-level grouping modules. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 11 (1997) 1306–1312
9. Brown, P.F., Della Pietra, S.A., Della Pietra, V.J., Mercer, R.L.: The Mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, Vol. 19, No. 2 (1993) 263–311
10. Cadez, I., Smyth, P.: Parameter estimation for inhomogeneous Markov random fields using PseudoLikelihood. University of California, Irvine (1998)
11. Carbonetto, P., de Freitas, N., Gustafson, P., Thompson, N.: Bayesian feature weighting for unsupervised learning, with application to object recognition. *Workshop on Artificial Intelligence and Statistics* (2003)
12. Dorkó, G., Schmid, C.: Selection of scale invariant neighborhoods for object class recognition. *Intl. Conf. Comp. Vision* (2003)
13. Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D.A.: Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. *European Conf. Comp. Vision* (2002)
14. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. *IEEE Conf. Comp. Vision and Pattern Recognition* (2003)
15. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision. *Intl. J. of Comp. Vision*, Vol. 40, No. 1 (2000) 23–47
16. Kumar, S., Hebert, H.: Discriminative Random Fields: a discriminative framework for contextual interaction in classification. *Intl. Conf. Comp. Vision* (2003)
17. Kumar, S., Hebert, H.: Discriminative Fields for modeling spatial dependencies in natural images. *Adv. in Neural Information Processing Systems*, Vol. 16 (2003)
18. Lowe, D.G.: Object recognition from local scale-invariant features. *Intl. Conf. Comp. Vision* (1999)
19. Murphy, K., Weiss, Y., Jordan, M.I.: Loopy belief propagation for approximate inference: an empirical study. *Conf. Uncertainty in Artificial Intelligence* (1999)
20. Seymour, L.: Parameter estimation and model selection in image analysis using Gibbs-Markov random fields. PhD thesis, U. of North Carolina, Chapel Hill (1993)
21. Mikolajczyk, K., Schmid, C.: A Performance evaluation of local descriptors. *IEEE Conf. Comp. Vision and Pattern Recognition* (2003)
22. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Conf. Comp. Vision and Pattern Recognition* (1997)
23. Teh, Y.W., Welling, M.: The Unified propagation and scaling algorithm. *Advances in Neural Information Processing Systems*, Vol. 14 (2001)

Reconstruction from Projections Using Grassmann Tensors

Richard I. Hartley¹ and Fred Schaffalitzky²

¹ Australian National University and National ICT Australia, Canberra

² Australian National University, Canberra.

Abstract. In this paper a general method is given for reconstruction of a set of feature points in an arbitrary dimensional projective space from their projections into lower dimensional spaces. The method extends the methods applied in the well-studied problem of reconstruction of a set of scene points in \mathcal{P}^3 given their projections in a set of images. In this case, the bifocal, trifocal and quadrifocal tensors are used to carry out this computation. It is shown that similar methods will apply in a much more general context, and hence may be applied to projections from \mathcal{P}^n to \mathcal{P}^m , which have been used in the analysis of dynamic scenes. For sufficiently many generic projections, reconstruction of the scene is shown to be unique up to projectivity, except in the case of projections onto one-dimensional image spaces (lines).

1 Introduction

The bifocal tensor (fundamental matrix), trifocal tensor and quadrifocal tensor have been much studied as a means of reconstructing a 3-dimensional scene from its projection in two, three or four images. It is well known that given sufficiently many point (or line) correspondences between the views, it is possible to compute the multiview tensor and subsequently extract from it the original projection matrices of the cameras, up to an unavoidable projective equivalence. There have been too many papers related to this to cite them all, and so we refer here only to the following papers: [3,2,4]. The methods previously given for extracting the projection matrices from the bifocal, trifocal and quadrifocal tensor have been quite different, and it was not clear that a general method exists.

In work involving the analysis of dynamic scenes, Wolf and Shashua ([11]) have considered projections from higher-dimensional projective spaces $\mathcal{P}^n \rightarrow \mathcal{P}^2$. They showed that such problems can also be studied in terms of tensors, and give some methods for working with these tensors. They do not, however give a general method for defining these tensors, or extracting the projection matrices afterwards. Neither do they consider projections into higher dimensional projective spaces.

At the other end of the scale, Quan and others ([7,1]) have studied projections between low-dimensional spaces, namely projections from \mathcal{P}^2 to \mathcal{P}^1 , and solve the reconstruction problem using a trifocal tensor. Quan shows ([6]) that in this case, there are two possible reconstructions.

This paper unifies all this previous work by showing that reconstruction from projections of \mathcal{P}^n into arbitrary dimensional projective spaces is always possible, and is almost always projectively unique. The method involves a generalization of the multiview tensors for projections $\mathcal{P}^3 \rightarrow \mathcal{P}^2$ (referred to subsequently as the “classical” tensors). The exception is the case where all the projections are onto one-dimensional projective spaces (lines). In this case, two reconstructions are always possible.

The reconstruction method described in this paper involves Grassmann tensors, which relate Grassmann coordinates¹ of linear subspaces in the image. The concept of Grassmann tensor was introduced by Triggs ([9,10]) to unify the classical bifocal, trifocal and quadrifocal tensors. The same formalism was taken up by Heyden in a paper exploring the multilinear matching relations ([5]). Triggs paper does not put any restriction on the dimensions of projective spaces considered, though he seems mainly to be concerned with the classical projection $\mathcal{P}^3 \rightarrow \mathcal{P}^2$, and point correspondences. Nevertheless, in [9] he observes that relations exist involving Grassmann coordinates of higher-dimensional subspaces, though he does not pursue the subject. In this paper, we consider this topic in detail, defining a general class of tensors. The main result of this paper, however, is that from any of these tensors the projection matrices may be retrieved, up to projective equivalence. This result does not appear in the papers of Triggs or Heyden.

The Grassmann tensor. We consider a sequence of projections from \mathcal{P}^n to \mathcal{P}^{m_i} , for $i = 1, \dots, r$. Thus, we do not assume that the image space always has the same dimension. For each i , we select integers α_i satisfying $1 \leq \alpha_i \leq m_i$ and $\sum_i \alpha_i = n + 1$. These values represent the codimension of linear subspaces to be specified in each of the image spaces. Thus, when $\alpha_i = m_i$, the linear subspace is a point (dimension 0), and when $\alpha_i = 1$, the linear subspace is a codimension-1 hyperplane. A set of linear subspaces with these dimensions are said to *correspond* when there exists at least one point \mathbf{X} in \mathcal{P}^n that maps via each projection to a point in the given linear subspace in the corresponding image space.

For instance, in the case of the classical trifocal tensor, we say that $\mathbf{x} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$ is a point-line-line correspondence if there exists a point \mathbf{X} in \mathcal{P}^3 that maps to \mathbf{x} in the first image, and to points on the lines \mathbf{l}' and \mathbf{l}'' in the other two images. The corresponding point and lines satisfy a relationship $\sum_{i,j,k} x^i l_j' l_k'' \mathcal{T}_i^{jk} = 0$ ([3]).

In the general case now being considering, there also exists a tensor relating the coordinates of a set of corresponding linear subspaces in the set of images. However, to assign coordinates to linear subspaces of arbitrary dimension, we need to use Grassmann coordinates (described later). Note that for points and lines in \mathcal{P}^2 , the Grassmann coordinates are nothing more than the homogeneous coordinates of the point or line. It is only when we consider image spaces of higher dimension that the correct generalization in terms of Grassmann coordinates

¹ Grassman coordinates, used in this paper, are also called Plücker coordinates by some authors.

becomes apparent. In this case, the tensor relates the Grassmann coordinates of the corresponding linear subspaces in each image. The relationship is of the form

$$\sum_{\sigma_1, \sigma_2, \dots, \sigma_r} |S_{\sigma_1}^1| |S_{\sigma_2}^2| \dots |S_{\sigma_r}^r| \mathcal{A}_{\sim \sigma_1 \sim \sigma_2 \dots \sim \sigma_r} = 0 \quad (1)$$

The notation $|S_{\sigma_i}^i|$ represents the σ_i -th Grassmann coordinate of the subspace S^i . Recall that S^i is a subspace of codimension α_i in \mathcal{P}^{m_i} . Consequently, the vector of Grassmann coordinates has dimension $C_{m_i+1}^{\alpha_i}$, and σ_i is an index into this Grassmann vector. The sum is over all combinations of Grassmann coordinates. The notation $\sim \sigma_i$ is to be read “not” σ_i . What this means is not made clear until later, but the reader may safely ignore the \sim , for it is only a notational convenience (or perhaps inconvenience). In fact $\mathcal{A}_{\sim \sigma_1 \sim \sigma_2 \dots \sim \sigma_r}$ represents a tensor indexed by the indices σ_i .² We refer to \mathcal{A} as a *Grassmann tensor*.

Computation of the Grassmann tensor. Given a correspondence between subspaces of codimension α_i in each \mathcal{P}^{m_i} , we obtain a single linear relationship between the elements of the Grassmann tensor. It is also possible to obtain relationships involving the tensor in the case of correspondences between subspaces of greater codimension than α_i . In the case of the trifocal tensor, a 3-point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}' \leftrightarrow \mathbf{x}''$ leads to four linear relations. These are obtained by choosing any two lines passing through \mathbf{x}' and any two lines passing through \mathbf{x}'' . Each choice of lines leads to a point-line-line correspondence, from each of which one obtains a linear relation. This same idea allows us to derive linear relations for Grassmann tensors in higher dimension, given a correspondence between subspaces of higher codimension. The exact number of correspondences generated in this way is not explored in this paper, though it is well understood in the $\mathcal{P}^3 \rightarrow \mathcal{P}^2$ case. In any case, given sufficiently many correspondences the Grassmann tensor may be computed linearly.

For clarification, it should be pointed out that for a set of projections $\mathcal{P}^n \rightarrow \mathcal{P}^{m_i}$, there may be many different tensors, depending on the choice of the sequence of codimensions $(\alpha_i, \alpha_2, \dots, \alpha_r)$. The only restrictions are that $1 \leq \alpha_i \leq m_i$ and $\sum_i \alpha_i = n + 1$. In the well-known case of the trifocal tensor, there are actually three different tensors depending on which of the three images is chosen to have the contravariant index. The three tensors have codimension sequences $(2, 1, 1)$, $(1, 2, 1)$ and $(1, 1, 2)$ respectively. In the general case, we call the sequence of codimensions $(\alpha_1, \alpha_2, \dots, \alpha_r)$ the *profile* of the corresponding tensor. Each such profile corresponds to a different tensor. If we are computing a tensor from point correspondences across several views, then it is necessary to choose in advance which profile to use, since any profile consistent with the dimensions of the image spaces can be used.

Extraction of projection matrices. Having computed a Grassmann tensor from a set of linear subspace correspondences, we now seek to extract the projection matrices. Ad-hoc techniques for computing the projections from multiview tensors have been proposed in the past, both for the standard case of $\mathcal{P}^3 \rightarrow \mathcal{P}^2$ as

² In some respects the \sim sign is analogous to the use of upper and lower indices in the classical tensor notation.

well as for higher dimensional cases ([11]). We now give a general procedure for doing this, and show that (at least for generic projections) the projection matrices are determined uniquely by a Grassmann tensor up to projective equivalence, **except** in the case where each $m_i = 1$. In this latter case, there will always be two non-equivalent solution, and indeed this represents a basic ambiguity for projective reconstruction from projections onto lines. This ambiguity persists however many point correspondences are involved. The two projective reconstructions are related to each other by a Cremona transform, which is a non-linear transformation of \mathcal{P}^n ([8]).

The method for computing the projection matrices given the tensor is related to the way a spanning set of vectors for a linear subspace is computed from the Grassmann coordinates. However, it is somewhat more involved. We make no claim that this method is optimum, or even robust in the presence of noise. In fact we are not concerned with noise at all. The present paper provides an existence and uniqueness proof for reconstruction rather than attempting to determine an optimal algorithm. As a deliberate choice, no experimental results will be reported.

1.1 Definition of the Tensors

A mapping from \mathcal{P}^n to \mathcal{P}^m is represented by a matrix of dimension $(m+1) \times (n+1)$, acting on homogeneous coordinates. We consider a set of r such mappings, where the i -th mapping is from \mathcal{P}^n to \mathcal{P}^{m_i} . Thus the dimension of the image of this mapping may be different in each case. The matrix representing the i -th mapping will be denoted by A^i .

We introduce the concept of an *ordered partition* of $n+1$. This is an ordered tuple of non-negative integers $(\alpha_1, \alpha_2, \dots, \alpha_r)$ that sum to $n+1$. We are interested in those partitions of n for which each α_i lies in the range 1 to m_i . We will show that for each such ordered partition, there exists an r -view tensor (where r is the length of the partition) relating the coordinates of matched codimension- α_i linear subspaces in r images.

Thus when $n = 3$ and each $m_i = 2$, the possible ordered partitions of $4 = 3+1$ are $(2, 2)$, $(2, 1, 1)$, $(1, 2, 1)$, $(1, 1, 2)$ and $(1, 1, 1, 1)$. These partitions correspond to the well-known multiview tensors for 2, 3 and 4 views. We see that there is a bi-focal tensor (the fundamental matrix) corresponding to the partition $(2, 2)$, three trifocal tensors corresponding to the three partitions of length 3, and one quadrfocal tensor.

We will call the ordered partition corresponding to a given tensor the *profile* of the tensor. How the tensor with a given profile is defined will now be explained.

Given $d+1$ points spanning a linear subspace of some projective space, we assemble the points as the columns of a matrix S . The linear subspace is simply the span of the columns of S and any point in this subspace can be written in the form $S\mathbf{v}$ for some suitable vector \mathbf{v} . We may speak of the matrix S as *representing* the subspace. The condition for a point \mathbf{X} in \mathcal{P}^n to map into the subspace under a mapping represented by A is that $A\mathbf{X} - S\mathbf{v} = 0$ for some \mathbf{v} .

Now, choose a set of linear subspaces each of codimension α_i in its projective space \mathcal{P}^{m_i} and let \mathbf{S}^i be the matrix representing the subspace. Suppose that there exists a point \mathbf{X} in \mathcal{P}^n that maps under each projection (represented by \mathbf{A}^i) to a point lying in the subspace \mathbf{S}^i . It will be shown that this condition implies a single constraint on the set of projection matrices \mathbf{A}^i .

The fact that this same \mathbf{X} projects into each of the subspaces may be written in one matrix equation as follows.

$$\begin{bmatrix} \mathbf{A}^1 & \mathbf{S}^1 \\ \mathbf{A}^2 & \mathbf{S}^2 \\ \vdots & \ddots \\ \mathbf{A}^r & \mathbf{S}^r \end{bmatrix} \begin{pmatrix} \mathbf{X} \\ -\mathbf{v}_1 \\ -\mathbf{v}_2 \\ \vdots \\ -\mathbf{v}_r \end{pmatrix} = \mathbf{0} . \quad (2)$$

Note that the matrix on the left is square. To check this: the number of rows is equal to $\sum_{i=1}^r (m_i + 1)$, whereas the number of columns is equal to

$$(n + 1) + \sum_{i=1}^r (m_i + 1 - \alpha_i) = \sum_{i=1}^r (m_i + 1) \quad \text{since} \quad \sum_{i=1}^r \alpha_i = (n + 1) .$$

In order for a non-zero solution to this set of equations to exist, it is necessary that the determinant of the matrix be zero. If the coordinates of the subspaces (the matrices \mathbf{S}^i) are given, then this provides a single constraint on the entries of the matrices \mathbf{A}^i . To understand the form of this constraint, we need to expand out this determinant, and to do that, we shall need to use Grassmann coordinates.

Grassmann coordinates. Given a matrix \mathbf{M} with q rows and p columns, where $p \leq q$, we define its *Grassmann coordinates* to be the sequence of determinants of all its $p \times p$ submatrices. It is a well known fact that the Grassmann coordinates of a matrix determine its column span. Alternatively, the Grassmann coordinates determine the matrix up to right-multiplication by a non-singular $p \times p$ matrix with unit determinant. Let σ represent a sequence of p distinct integers in the range 1 to q , in ascending order. Let $|\mathbf{M}_\sigma|$ represent the determinant of the matrix that consists of the rows of \mathbf{M} specified by the sequence σ . Then the values $|\mathbf{M}_\sigma|$, as σ ranges over all such sequences, are the Grassmann coordinates of the matrix.

Now, given such a sequence σ indexing the rows of a matrix, let $\sim \sigma$ represent the sequence consisting of those integers not in σ . and define $\text{sign}(\sigma)$ to be $+1$ or -1 depending on whether the concatenated sequence $\sigma \sim \sigma$ is an even or odd permutation.³ Thus, for example the sequence 125 has $\text{sign} +1$, since 12534 is an even permutation.

Given a square matrix divided into two blocks, for instance $[\mathbf{A}|\mathbf{B}]$, its determinant may be expressed in terms of the Grassmann coordinates of \mathbf{A} and \mathbf{B} . In particular

$$|\mathbf{A}|\mathbf{B}| = \sum_{\sigma} \text{sign}(\sigma) \|\mathbf{A}_\sigma\| \|\mathbf{B}_{\sim \sigma}\|$$

³ A permutation is called even or odd, according to whether it is the product of an even or odd number of pairwise swaps.

where the sum is over all ascending sequences σ of length equal to the number of columns of \mathbf{A} . The particular case where \mathbf{A} consists of a single column is just the familiar cofactor expansion of the determinant.

Using this factorization, one may derive a precise formula for the determinant of the matrix on the left of (2), namely

$$\pm \sum_{\sigma_1, \sigma_2, \dots, \sigma_r} \text{sign}(\sigma_1) \dots \text{sign}(\sigma_r) |\mathbf{A}_{\sim \sigma_1 \sim \sigma_2 \dots \sim \sigma_r}| |\mathbf{S}_{\sigma_1}^1| |\mathbf{S}_{\sigma_2}^2| \dots |\mathbf{S}_{\sigma_r}^r|. \quad (3)$$

In this formula, each σ_i is an ordered sequence of integers in the range 1 to $m_i + 1$, the length of the sequence being equal to the dimension of the subspace \mathbf{S}^i . Further, $|\mathbf{A}_{\sim \sigma_1 \sim \sigma_2 \dots \sim \sigma_r}|$ is the determinant of the matrix obtained by selecting the rows indexed by $\sim \sigma_i$ (that is, omitting the rows indexed by σ_i) from each \mathbf{A}^i . The overall sign (whether + or -) does not concern us. The set of values

$$\mathcal{A}_{\sim \sigma_1 \sim \sigma_2 \dots \sim \sigma_r} = \text{sign}(\sigma_1) \dots \text{sign}(\sigma_r) |\mathbf{A}_{\sim \sigma_1 \sim \sigma_2 \dots \sim \sigma_r}|$$

forms an r dimensional array whose elements are (up to sign) minors of the matrix \mathbf{A} obtained by stacking the projection matrices \mathbf{A}^i . The only minors are ones corresponding to submatrices of \mathbf{A} , in which α_i rows are chosen from each \mathbf{A}^i . Recalling that the sequence $(\alpha_1, \dots, \alpha_r)$ in which $\sum_{i=1}^r \alpha_i = n + 1$ is called a *profile*, we will call the array $\mathcal{A}_{\sim \sigma_1 \sim \sigma_2 \dots \sim \sigma_r}$ the *Grassmann tensor* corresponding to the profile $(\alpha_1, \dots, \alpha_r)$.

The tensor \mathcal{A} gives a linear relationship between the Grassmann coordinates of linear subspaces defined in each of the image spaces \mathcal{P}^{m_i} :

$$\sum_{\sigma_1, \sigma_2, \dots, \sigma_r} \mathcal{A}_{\sim \sigma_1 \sim \sigma_2 \dots \sim \sigma_r} |\mathbf{S}_{\sigma_1}^1| |\mathbf{S}_{\sigma_2}^2| \dots |\mathbf{S}_{\sigma_r}^r| = 0. \quad (4)$$

This relationship generalizes the classical bifocal and trifocal relations ([3]). The classical tensors involve relations between point and line coordinates in \mathcal{P}^2 . However, the Grassmann coordinates of a single point (a 0-dimensional linear space) are simply the homogeneous coordinates of the point. Similarly, for a line in \mathcal{P}^2 , the Grassmann coordinates are the same as the homogeneous coordinates, except for sign.

Given a change of coordinates in some of the image spaces \mathcal{P}^{m_i} , the tensor \mathcal{A} does not in general transform strictly as a contravariant or covariant tensor. Rather, it transforms according to the inverse of the corresponding transformation of Grassmann coordinates induced by the change of coordinates. This map is the $m_i + 1 - \alpha_i$ -fold exterior product mapping induced by the coordinate change.

2 Solving for the Projection Matrices

We now consider the problem of determining the projection matrices from a Grassmann tensor. As in the standard case of 3D reconstruction from uncalibrated image measurements, we can not expect to determine the projection matrices more exactly than up to projectivity. In addition, since the projection

matrices are homogeneous objects, their overall scale is indeterminate. Thus, we make the following definition:

Definition 1. *Two sequences of projection matrices $(\mathbf{A}^1, \dots, \mathbf{A}^r)$ and $(\hat{\mathbf{A}}^1, \dots, \hat{\mathbf{A}}^r)$ are projectively equivalent if there exists an invertible matrix \mathbf{H} as well as scalars λ_i such that $\hat{\mathbf{A}}^i = \lambda_i \mathbf{A}^i \mathbf{H}$ for all i .*

Now, let \mathbf{A} be formed by stacking all the \mathbf{A}^i on top of each other, resulting in a matrix of dimension $(\sum_{i=1}^r (m_i + 1)) \times (n + 1)$. We accordingly associate the matrices \mathbf{A}^i with successive vertically stacked blocks of the matrix \mathbf{A} . Corresponding to definition 1, we may define an equivalence relation on matrices with this block structure, as follows.

Definition 2. *Two matrices \mathbf{A} and $\hat{\mathbf{A}}$ made up of blocks \mathbf{A}^i and $\hat{\mathbf{A}}^i$ of dimension $(m_i + 1) \times (n + 1)$ are block projectively equivalent if there exists an invertible $(n+1) \times (n+1)$ matrix \mathbf{H} , and scalar matrices $\lambda_i \mathbf{I}_{m_i}$ of dimension $(m_i + 1) \times (m_i + 1)$ such that*

$$\hat{\mathbf{A}} = \text{diag}(\lambda_1 \mathbf{I}_1, \dots, \lambda_r \mathbf{I}_r) \mathbf{A} \mathbf{H} .$$

It is easily seen that this definition is equivalent to the projective equivalence of the sequences of matrices $(\mathbf{A}^1, \dots, \mathbf{A}^r)$ and $(\hat{\mathbf{A}}^1, \dots, \hat{\mathbf{A}}^r)$ as stated in definition 1. It is evident that this is an equivalence relation on matrices with this given block structure.

2.1 Partitions and Determinants

Now, we assume that there are sufficiently many such projections that $\sum_{i=1}^r m_i \geq n$. Let $(\alpha_1, \dots, \alpha_r)$ be an ordered partition of $(n + 1)$ with the property that $1 \leq \alpha_i \leq m_i$. We may form square matrices of dimension $(n+1) \times (n+1)$ by selecting exactly α_i rows from each matrix \mathbf{A}^i . We may then take the determinant of such a square matrix. Of course, we may select α_i rows from each \mathbf{A}^i in many different ways – to be exact, there are $\prod_{i=1}^r C_{m_i+1}^{\alpha_i}$ ways of doing this, and that many such subdeterminants of \mathbf{A} corresponding to the given partition.

Before giving the main theorem, we state our assumption of genericity. All projections from \mathcal{P}^n to \mathcal{P}^m are assumed to be “generic”, which means in effect that improbable special cases are ruled out. Any polynomial expression in the coordinates of the matrix representation of the projections, or related points may be assumed to be non-zero, unless it is always zero. Thus the results we prove will hold, except on a set of measure zero. We now state the main theorem of this part of the paper.

Theorem 1. *Let \mathbf{A} be a generic matrix with blocks \mathbf{A}^i ; $i = 1, \dots, r$ of dimension $(m_i + 1) \times (n + 1)$, and let $(\alpha_1, \dots, \alpha_r)$ be any fixed ordered partition of $n + 1$. If at least one m_i is greater than one, then the matrix \mathbf{A} is determined up to block projective equivalence by the collection of all its minors, chosen with α_i rows from each \mathbf{A}^i . If all $m_i = 1$, then there are two equivalence classes of solutions.*

We refer to the partition $(\alpha_1, \dots, \alpha_r)$ as the *profile* of the minors. Thus, the theorem states that the matrix \mathbf{A} is determined up to projective equivalence by its collection of minors with a given fixed profile.

Proof. We would like to give a more leisurely proof, with examples, but are prevented by lack of space. The proof given below is complete, but telegraphic. Let \mathbf{A} and $\hat{\mathbf{A}}$ be two matrices with corresponding blocks \mathbf{A}^i and $\hat{\mathbf{A}}^i$, each of which gives rise to the same collection of minors. Our goal is to show that the two matrices are block projective-equivalent, which means that $\hat{\mathbf{A}}^i = \lambda_i \mathbf{A}^i \mathbf{H}$ for some choice of \mathbf{H} and λ_i . The strategy of the proof is to apply a sequence of transformations to \mathbf{A} (and to $\hat{\mathbf{A}}$), each transformation replacing \mathbf{A} by a projectively equivalent matrix, until eventually \mathbf{A} and $\hat{\mathbf{A}}$ become identical. This will demonstrate the projective equivalence of the original matrices.

By assumption, there exists at least one non-zero minor, and without loss of generality (by rearranging rows if required), this may be chosen to belong to the submatrix of \mathbf{A} in which the *first* α_i rows are chosen from each \mathbf{A}^i . Let this submatrix be denoted by \mathbf{G} . Choosing $\mathbf{H} = \mathbf{G}^{-1}$, we may replace \mathbf{A} by an equivalent matrix \mathbf{AH} in which the matrix \mathbf{G} is replaced by the identity matrix. Doing the same thing to $\hat{\mathbf{A}}$, we may assume that both \mathbf{A} and $\hat{\mathbf{A}}$ have this simple form.

After this transformation, the form of the matrix \mathbf{A} is somewhat simplified. The first α_i rows from each block are known, consisting of zeros, except for one unit element in each such row. We refer to these rows of \mathbf{A} as the *reduced* rows. The elements of the remaining rows of \mathbf{A} are still to be determined. We show that they can be determined (up to block projective equivalence) from other minors of the matrix.

We consider a finer block decomposition of the matrix \mathbf{A} into blocks indexed by (i, j) where the block \mathbf{A}^{ij} has dimension $(m_i + 1) \times \alpha_j$ as shown:

$$\begin{bmatrix} \mathbf{A}^{11} & \dots & \mathbf{A}^{1r} \\ \vdots & \ddots & \vdots \\ \mathbf{A}^{r1} & \dots & \mathbf{A}^{rr} \end{bmatrix}. \quad (5)$$

The first α_i rows of each such \mathbf{A}^{ij} are reduced, so

$$\mathbf{A}^{ii} = \begin{bmatrix} \mathbf{I} \\ \mathbf{B}^{ii} \end{bmatrix} \quad \text{and} \quad \mathbf{A}^{ij} = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}^{ij} \end{bmatrix} \quad \text{for } i \neq j$$

The reduced rows of \mathbf{A} form an identity matrix, having unit determinant. Let \mathbf{B} be the matrix obtained from \mathbf{A} by removing the reduced rows. Then \mathbf{B} has the same type of block structure as \mathbf{A} . We investigate the relationship between minors⁴ of \mathbf{A} and those of \mathbf{B} .

Consider a submatrix of \mathbf{A} chosen according to a given profile $(\alpha_1, \dots, \alpha_r)$. Some of the rows of this submatrix will be rows of \mathbf{B} , while other will be chosen

⁴ For brevity, when we speak of the minors of \mathbf{A} , we mean those chosen according to the given profile, with α_i rows from each \mathbf{A}^i .

from among the reduced rows of \mathbf{A} . A reduced row is one in which there is one unit (1) entry and the rest are zero. In computing the determinant, we may strike out any reduced rows, as well as the columns containing the unit entries, resulting in a smaller matrix containing only elements belonging to rows from \mathbf{B} . The columns that remain are ones that did not have a 1 in any reduced row of the chosen submatrix. Here is the key observation: if a row is chosen from the i -th block of rows $[\mathbf{B}^{i1} \dots \mathbf{B}^{ir}]$ then some reduced row from the same numbered blocks $[\mathbf{A}^{i1} \dots \mathbf{A}^{ir}]$ must be absent. Such a row has its unit element in the block \mathbf{A}^{ii} , but this row is not present in the chosen submatrix. The corresponding column, belonging to the i -th block of columns, must therefore “survive” when rows and columns corresponding to the reduced rows are struck out. This means:

The minors of \mathbf{A} are in one-to-one correspondence with (and equal up to sign to) the minors of \mathbf{B} chosen in the following manner: β_i rows are chosen from the i -th block of rows of $[\mathbf{B}^{i1} \dots \mathbf{B}^{ir}]$ and β_i columns from the i -th block of columns, containing the blocks $\mathbf{B}^{i1} \dots \mathbf{B}^{ri}$. Here the β_i are integers in the range $0 \leq \beta_i \leq \alpha_i$.

Such minors of \mathbf{B} will be called “symmetrically chosen” minors. The minors of \mathbf{A} and \mathbf{B} are equal only up to sign, because of the order of the rows, but the sign correspondence is well determined, so that if one knows the values of the minors of \mathbf{A} , then the symmetrically chosen minors of \mathbf{B} are also known. We will show that \mathbf{B} is determined by its symmetrically chosen minors, and hence by the minors of \mathbf{A} . Therefore, knowing the minors of \mathbf{A} , we know \mathbf{B} and hence \mathbf{A} , up to projective equivalence. This would complete the proof.

The exact truth is slightly more complicated. We define a different type of equivalence relation on block matrices of the form $\mathbf{B} = [\mathbf{B}^{ij}]$, where $i, j = 1, \dots, r$.

Definition 3. *Two matrices $\mathbf{B} = [\mathbf{B}^{ij}]$ and $\hat{\mathbf{B}} = [\hat{\mathbf{B}}^{ij}]$ will be called bilinearly equivalent if there exist non-zero scalars λ_i such that $\hat{\mathbf{B}}^{ij} = \lambda_i \lambda_j^{-1} \mathbf{B}^{ij}$.*

The truth is that the symmetrically chosen minors of \mathbf{B} determine \mathbf{B} up to bilinear equivalence. This is sufficient, however, since if \mathbf{B} and $\hat{\mathbf{B}}$ are bilinearly equivalent, then the corresponding matrices \mathbf{A}^i and $\hat{\mathbf{A}}^i$ are projectively equivalent, which is all we need. This is true because

$$\hat{\mathbf{A}}^i = \lambda_i \mathbf{A}^i \text{diag}(\lambda_1^{-1} \mathbf{I}_{\alpha_1}, \dots, \lambda_r^{-1} \mathbf{I}_{\alpha_r})$$

follows from the fact that $\hat{\mathbf{B}}^{ij} = \lambda_i \lambda_j^{-1} \mathbf{B}^{ij}$.

The proof of Theorem 1 will be completed therefore by proving the following lemma.

Lemma 1. *A matrix $\mathbf{B} = [\mathbf{B}^{ij}]$ is determined up to bilinear equivalence by its collection of symmetrically chosen minors.*

In fact, it will be sufficient only to consider only 3×3 minors, or 2×2 minors in the two-view case. The proof will proceed in three steps.

1. The 1×1 minors determine the elements of the diagonal blocks \mathbf{B}^{ii} .
2. The 2×2 minors determine symmetrically opposite pairs of blocks \mathbf{B}^{ij} and \mathbf{B}^{ji} up to a pair of inverse scalar multiples.
3. The 3×3 minors determine consistent scale factors.

Step 1 - the 1×1 minors. The 1×1 symmetrically chosen minors are nothing other than the elements of the diagonal blocks \mathbf{B}^{ii} , and hence these 1×1 minors determine the diagonal blocks.

Step 2 - the 2×2 minors. A 2×2 symmetrically chosen minor will be of the form $[a \ b; c \ d]$, where a and d are from diagonal blocks \mathbf{B}^{ii} and \mathbf{B}^{jj} , and hence are known from the previous step. Elements b and c are from the symmetrically opposite blocks \mathbf{B}^{ij} and \mathbf{B}^{ji} . Since the determinant is $ad - bc$ with ad known, we may obtain the value of bc from the value of the 2×2 minor. In fact, by choosing the right minor, we can determine the product of any two elements chosen from symmetrically opposite blocks such as \mathbf{B}^{ij} and \mathbf{B}^{ji} .

Let \mathbf{b} be the vector consisting of all elements from the block \mathbf{B}^{ij} and \mathbf{c} be the vector of elements of \mathbf{B}^{ji} . Then the set of all products v_{rs} of elements from blocks \mathbf{B}^{ij} and \mathbf{B}^{ji} can be determined and written $b_r c_s = v_{rs}$. This means that the values v_{rs} form a rank-1 matrix that factors as $\mathbf{V} = \mathbf{bc}^\top$. The factorization is easily carried out using the Singular Value Decomposition, or some more simple method⁵.

Solution of the equations $b_r c_s = w_{rs}$ is only possible up to an indeterminate scale factor. Thus, we may multiply each b_r by λ and c_s by λ^{-1} with the same result, but this is the only possible ambiguity. The result of this is that one may determine the blocks \mathbf{B}^{ij} and \mathbf{B}^{ji} of the matrix \mathbf{B} up to multiplication by inverse scalar factors.

Let $\mathbf{B} = [\mathbf{B}^{ij}]$ and $\hat{\mathbf{B}} = [\hat{\mathbf{B}}^{ij}]$ be two sets of matrices having the same collection of symmetrically chosen minors. Our goal is to show that $\hat{\mathbf{B}}^{ij} = \lambda_i \lambda_j^{-1} \mathbf{B}^{ij}$ for all i, j . On the other hand, what we have shown so far is that there exist non-zero constants μ_{ij} with $\mu_{ii} = 1$ and $\mu_{ij} = \mu_{ji}^{-1}$ such that $\hat{\mathbf{B}}^{ij} = \mu_{ij} \mathbf{B}^{ij}$. It remains to show that μ_{ij} can be written as $\lambda_i \lambda_j^{-1}$. At this point, we modify the matrix \mathbf{B} by multiply each block \mathbf{B}^{ij} by $\mu_{1i} \mu_{1j}^{-1}$. This operation transforms \mathbf{B} to another matrix that is bilinearly equivalent to it, and it is sufficient to prove that the new \mathbf{B} thus obtained is equivalent to $\hat{\mathbf{B}}$. Note however that because of this modification to the matrix \mathbf{B} , the first row block and column block of \mathbf{B} and $\hat{\mathbf{B}}$ are identical. Thus, in particular $\hat{\mathbf{B}}^{ij} = \mu_{ij} \mathbf{B}^{ij}$, and $\mu_{i1} = \mu_{1i} = 1$ for all i .

Step 3 - consistent choice of scale factors λ . The proof will now be completed by proving that $\mu_{ij} = 1$ for all i, j .

Consider allowable 3×3 subdeterminants of \mathbf{B} , in which one row is taken from the first row block, and one row each from each of two other row blocks. Corresponding columns are chosen from the corresponding blocks. The submatrix of \mathbf{B} is

$$\begin{bmatrix} a & b & c \\ d & e & \mu f \\ g & \mu^{-1} h & k \end{bmatrix} \quad (6)$$

and the submatrix of $\hat{\mathbf{B}}$ is the same in which $\mu = 1$. Equating the two determinants gives an equation of the form $A\mu + B + C\mu^{-1} = A + B + C$ for constants

⁵ The only thing that can go wrong here is that all w_{ij} are zero, but this a non-generic case.

A , B and C . Multiplying by μ gives a quadratic equation with two solutions: $\mu = 1$ and $\mu = C/A$. In terms of the entries of the matrix, the second solution is $\mu = (dhc)/(bfg)$. Thus, there are two possible solutions for μ . However, in most situations we may obtain a second equation for μ which will also have two solutions, but only the solution $\mu = 1$ will be common to both equations, and hence a solution to the complete system.

To see this, we need to make an assumption that the first projection matrix $\mathbf{A}^1 = [A^{11} \dots A^{1r}]$ has more than two rows – its dimension is $m_1 + 1 \geq 3$. This is possible without loss of generality provided that there exists at least one projection matrix with at least three rows, for it may be chosen as the first. The number of rows chosen from \mathbf{A}^1 is α_1 which is in the range $1 \leq \alpha_1 \leq m_1$. Now, suppose that the rows and columns of (6) are chosen from the row and column blocks numbered 1, i and j . Thus, the entries of (6) are drawn from the block matrix

$$\begin{bmatrix} B^{11} & B^{1i} & B^{1j} \\ B^{i1} & B^{ii} & B^{ij} \\ B^{j1} & B^{ji} & B^{jj} \end{bmatrix}. \quad (7)$$

Now, the dimension of the matrix B^{ij} is $(m_i + 1 - \alpha_i) \times \alpha_j$. Specifically, B^{i1} has dimension $(m_i + 1 - \alpha_i) \times \alpha_1$, and B^{1j} has dimension $(m_1 + 1 - \alpha_1) \times \alpha_j$. However, since $1 \leq \alpha_1 \leq m_1$ and $m_1 > 1$, it must follow that either $\alpha_1 > 1$ or $m_1 + 1 - \alpha_1 > 1$. Thus, either B^{i1} has at least two columns, or B^{1j} has at least two rows. In either case, there is more than one way of selecting rows and columns from (7) to obtain a submatrix of the form (6). Each such choice will give a different equation for μ . The solution $\mu = 1$ will be common to both equations whereas the second solution $\mu = (dhc)/(bfg)$ will be different for the two cases, since generically the entries of the matrix (6) will be different for the different choices of rows and columns.

This completes the proof, since we have shown that the only value of μ_{ij} that is consistent with the assumed equality of all the allowable minors of \mathbf{B} and $\hat{\mathbf{B}}$ is that $\mu_{ij} = 1$ for all i, j . Hence, $\mathbf{B} = \hat{\mathbf{B}}$.

2.2 Two Solutions in Minimal Case

It was seen that the case where all projection matrices have only two rows is a special case in which we can not find two equations for each value μ_{ij} . In such a case it is possible that there will be two possible solutions for matrices with the same set of minors. We will investigate this further, and show that in this case, generically there are indeed two solutions.

Thus, let the projection matrices \mathbf{A}^i each have dimension $2 \times (n + 1)$, representing a projection from an n -dimensional projective space onto a projective line. In order for us to form square submatrices by choosing one row from each such \mathbf{A}^i there must be exactly $n + 1$ such projections. Thus, \mathbf{A} has dimension $2(n + 1) \times (n + 1)$ and each $\alpha_i = 1$.

With the same argument as before, we may assume that the first rows of each of the \mathbf{A}^i form an identity matrix of dimension $(n + 1) \times (n + 1)$. Deleting

these rows, we obtain an $(n+1) \times (n+1)$ matrix B . In this case, each B^{ij} consists of a single element. We consider symmetrically chosen submatrices of B . In this case, the symmetrically chosen submatrices are those for which the indices of the selected rows and columns are the same. Such a submatrix is chosen by selecting the rows and columns numbered by a sequence of indices (i_1, i_2, \dots, i_r) . The key observation here is that the minors of B and its transpose B^T corresponding to a given sequence of indices are the same, because the determinant of a matrix and its transform are the same. In other words, it is impossible to distinguish between the matrix B and its transpose on the basis of such symmetrically chosen minors.

Referring this back to the original projection matrices we obtain two matrices A and \hat{A} which can not be distinguished by their minors with profile $(1, \dots, 1)$. We may write a specific example as follows: Let

$$A^1 = \begin{bmatrix} 1 & 0 & 0 \\ a & b & c \end{bmatrix} \quad ; \quad A^2 = \begin{bmatrix} 0 & 1 & 0 \\ d & e & f \end{bmatrix} \quad ; \quad A^3 = \begin{bmatrix} 0 & 0 & 1 \\ g & h & j \end{bmatrix}$$

and

$$\hat{A}^1 = \begin{bmatrix} 1 & 0 & 0 \\ a & d & g \end{bmatrix} \quad ; \quad \hat{A}^2 = \begin{bmatrix} 0 & 1 & 0 \\ b & e & h \end{bmatrix} \quad ; \quad \hat{A}^3 = \begin{bmatrix} 0 & 0 & 1 \\ c & f & j \end{bmatrix} .$$

The two matrices A and \hat{A} corresponding to these projection matrices can not be distinguished based on the eight minors formed by choosing one row from each A^i , or respectively \hat{A}^i . In terms of tensors, this means that the “trifocal tensors” corresponding to these triples of cameras are the same. Geometrically, this means that there are two possible reconstructions of a (planar) scene based on its projection onto three lines in the plane. The 1-dimensional trifocal tensor was studied by Quan and others in ([7,1]). The observation that there were two solutions in the case of projections from \mathcal{P}^2 to \mathcal{P}^1 was made in [6]. The ambiguity holds in higher dimensions also, as the above argument shows. Specifically, the tensor (collection of minors) corresponding to $n+1$ projections from \mathcal{P}^n onto a projective line determines the set of projection matrices only up to 2-fold projective ambiguity. Consequently there are always two reconstructions possible from the projection of \mathcal{P}^n onto $(n+1)$ projective lines.

Are there generically only two solutions? In the case where each $m_i = 1$, it may seem possible that there are more than two possibilities (up to bilinear equivalence) for the matrix B based on its set of minors. However, this is not possible. If we assume that two matrices B and \hat{B} have the same set of symmetric minors, then by carrying through the previous arguments, we find that B may be replaced by an equivalent matrix for which B and \hat{B} have the same first row. In addition, there exist constants μ_{ij} such that $\mu_{ij} = \mu_{ji}^{-1}$ and $B^{ij} = \mu_{ij} \hat{B}^{ij}$. By considering 3×3 symmetric minors containing the first row and column as in the proof of lemma 1 we obtain a single quadratic equation for each of the constants μ_{ij} . There are two choices. For each (i, j) we may choose $\mu_{ij} = 1$, or else we must choose the other non-unit solution at each position. It may be seen that once one value μ_{ij} with $i, j > 1$ is chosen to equal 1, then they all must be. The

details are tedious, and omitted. The solution in which μ_{ij} is taken to be the non-unit solution for each i, j may be verified to be equivalent (under bilinear equivalence) to the transposed solution in which $\mathbf{B} = \hat{\mathbf{B}}^\top$.

3 Conclusion

The classical multiview tensor extend to higher dimensions, and allow reconstruction of the scene from projections in any dimension. The solution is unique except in the case of projections onto lines. The work of Wolf and Shasha shows the importance of higher dimensional projections, and provides a potential application for this work, at least in proving the feasibility of a solution.

References

1. O. D. Faugeras, L. Quan, and P. Sturm. Self-calibration of a 1D projective camera and its application to the self-calibration of a 2D projective camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1179–1185, October 2000.
2. R. I. Hartley. Computation of the quadrifocal tensor. In *Proc. 5th European Conference on Computer Vision, Freiburg, Germany*, LNCS 1406, pages 20–35. Springer-Verlag, 1998.
3. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
4. A. Heyden. Reduced multilinear constraints: Theory and experiments. *International Journal of Computer Vision*, 30(1):5–26, 1998.
5. A. Heyden. Tensorial properties of multilinear constraints. *Mathematical Methods in the Applied Sciences*, 23:169–202, 2000.
6. L. Quan. Two-way ambiguity in 2d projective reconstruction from three uncalibrated 1d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):212–216, 2001.
7. L. Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):834–845, August 1997.
8. J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry*. Oxford University Press, 1979.
9. W. Triggs. The geometry of projective reconstruction i: Matching constraints and the joint image. *Unpublished: Available on Bill Triggs's web-site*, 1995.
10. W. Triggs. Matching constraints and the joint image. In E. Grimson, editor, *Proc. 5th International Conference on Computer Vision, Boston*, pages 338 – 343, Cambridge, MA, June 1995.
11. L. Wolf and A. Shashua. On projection matrices $\mathcal{P}^k \rightarrow \mathcal{P}^2, k = 3, \dots, 6$, and their applications in computer vision. *International Journal of Computer Vision*, 48(1):53–67, 2002.

Co-operative Multi-target Tracking and Classification

Pankaj Kumar¹, Surendra Ranganath², Kuntal Sengupta³, and Huang Weimin¹

¹ Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613,
kumar@i2r.a-star.edu.sg, wnhuang@i2r.a-star.edu.sg,

² National University of Singapore, 4 Engineering Drive 3 Singapore 117576,
elsesr@nus.edu.sg

³ AuthenTec, Inc. Post Office Box 2719, Melbourne, Florida 32902-2719
kuntal.sengupta@authentec.com

Abstract. This paper describes a real-time system for multi-target tracking and classification in image sequences from a single stationary camera. Several targets can be tracked simultaneously in spite of splits and merges amongst the foreground objects and presence of clutter in the segmentation results. In results we show tracking of upto 17 targets simultaneously. The algorithm combines Kalman filter-based motion and shape tracking with an efficient pattern matching algorithm. The latter facilitates the use of a dynamic programming strategy to efficiently solve the data association problem in presence of multiple splits and merges. The system is fully automatic and requires no manual input of any kind for initialization of tracking. The initialization for tracking is done using attributed graphs. The algorithm gives stable and noise free track initialization. The image based tracking results are used as inputs to a Bayesian network based classifier to classify the targets into different categories. After classification a simple 3D model for each class is used along with camera calibration to obtain 3D tracking results for the targets. We present results on a large number of real world image sequences, and accurate 3D tracking results compared with the readings from the speedometer of the vehicle. The complete tracking system including segmentation of moving targets works at about 25Hz for 352×288 resolution color images on a 2.8 GHz pentium-4 desktop.

1 Introduction

This paper address several problems of tracking and classifying multiple targets in real-time, which can be used for behavior analysis of the moving targets. Several new ideas have been developed to solve the problem of Multi-Target Tracking (MTT) in 3D under the following assumptions: 1. Image sequences are obtained from a single stationary camera looking down into the scene. 2. The targets are moving on a ground plane and some 3D measurements on the ground and their corresponding locations in the image are available for camera calibration. In this paper the problem of MTT is formulated as an optimal feature estimation and data association problem, which has been the usual paradigm

for MTT in radar community [1][2]. To obtain good classification results target tracking has to be accurate. The knowledge of object type can improve the tracking results. In this paper these two ideas have been combined to get accurate classification and 3D tracking results. First the issue of efficiently handling splits and merges in the segmentation of the targets so that tracking can continue even when targets split and merge and there are large number of targets in the Field of View (FOV). Second a new target track initialization algorithm is introduced which ensures accurate and stable initialization of the trackers which is usually required by many tracking algorithms. The third contribution is a new co-operative tracking-classifying-tracking algorithm. The results of image based 2D tracking are used to classify the target into different categories using a Bayesian network. This allows using a representative 3D model for each category to compute the 3D tracking results of the targets.

The paper is organized as follows: Related works and their differences to our work is discussed in Section 2. In Section 3 target modelling and pattern matching algorithm which facilitates the use of a dynamic programming (DP) strategy to efficiently compute the data association of targets in the presence of multiple splits and merges is discussed. Section 4 briefly discusses Kalman filter based motion and shape tracking. The new algorithm for automatic initialization of target tracking is discussed in Section 5. In Section 6 target classification based on Bayesian network is discussed. Section 7 explains the 3D models of the different classes and the camera calibration method used to obtain 3D tracking results from 2D tracking. Finally results and conclusions are presented in Sections 8 and 9.

2 Related Work

Paragios and Deriche [3] considered the problem of simultaneously tracking several non-rigid targets. The motion parameters of the targets were estimated using a coupled front propagation model, which integrates boundary and region-based information. McKenna *et al.*'s [4] work on tracking groups of people performs tracking at three levels of abstraction: regions, people, and groups. People are tracked through mutual occlusions as they form groups and separate from one another. Strong use of color information is made to disambiguate occlusion and to provide qualitative estimates of depth ordering and position during occlusion. Javed and Shah [5] presented an automated surveillance system where the objects were tracked and classified into different categories with a new feature, "Recurrent Motion Image" (RMI). The tracking discussed in [5] is based on region correspondence matching which may fail when there are large number of similar targets undergoing merges and splits. Haritaoglu *et al.* [6][7] proposed a system that combines shape analysis and statistical techniques to track people and their parts in an outdoor environment. To handle interactions amongst the tracked people, they used a generic human model tuned to each target's specific details to resolve the ambiguities. To track objects in $2\frac{1}{2}$ D they used stereo camera. In our approach 3D tracking results have been obtained using a single camera.

Medioni *et.al.* [8] proposed an approach similar to Reid's Multiple Hypothesis Tracking (MHT) [9][10]. They use attributed graph matching for creating new target tracks and tracking them. This approach is more advanced than MHT as it can handle cases where a target gives rise to multiple measurements because of splitting. The solution for MTT proposed in this paper can simultaneously handle both splitting and merging of targets in colored images. Tao *et al.* in [11] proposed dynamic motion layer based approach for tracking persons and vehicles in image sequences. Initialization of the tracker relies on blob detection. The system runs at 5 Hz for four moving objects in the scene. They show tracking results for 4 to 5 objects in the FOV. We show tracking results for 10-17 targets in the FOV at 25 Hz.

3 Feature Extraction and Pattern Matching

We use our active background modelling and foreground segmentation scheme to segment moving foreground objects in the Field of View (FOV) [12]. Another foreground segmentation technique which can be used is [13]. The foreground regions are enclosed within their convex hulls to remove concavities. If there are small connected regions lying within the convex hull of a larger connected region then the smaller regions are ignored and only the larger region is considered. The convex hulls are approximated by an ellipse using the algorithm given in [14]. The measurements obtained for each foreground region in a frame is called a Segmented Patch (*SP*) and its features are:

1. Centroid of the ellipse, X_c .
2. J angularly equidistant control points X_1, X_2, \dots, X_J on the ellipse.
3. The normalized, I bin histograms of the Y, Cr, Cb channels of the SP , H_1, H_2, \dots, H_I .

The o^{th} SP of a frame is represented as:

$$C^o = c_{X_c}^o, c_{X_1}^o, c_{X_2}^o, \dots, c_{X_J}^o, c_{H_1}^o, c_{H_2}^o, \dots, c_{H_I}^o. \quad (1)$$

The targets being tracked by the Kalman filter have same representation as the measurements with some extra features like velocity of the centroid $b_{V_c}^n$ and a parameter to measure change of shape b_s^n . The n^{th} target and its features are represented as:

$$B^n = b_{X_c}^n, b_{X_1}^n, b_{X_2}^n, \dots, b_{X_J}^n, b_{H_1}^n, b_{H_2}^n, \dots, b_{H_I}^n, b_{V_c}^n, b_s^n \quad (2)$$

3.1 Match Measures

Three match measures D_S , D_X and D_H are discussed here for matching targets with SP s based on shape and color information. The control points of the n^{th} target B^n are $b_{X_1}^n, b_{X_2}^n, \dots, b_{X_J}^n$. These control points form polygon $Poly_{B^n}$ and enclose area A_{B^n} . Similarly, the control points of the o^{th} SP , C^o form polygon $Poly_{C^o}$ and enclose area A_{C^o} . $(A_{B^n} \cap A_{C^o})$ is the common area between

the polygons $Poly_{B^n}$ and $Poly_{C^o}$. The match measures D_S and D_X used for matching shape of $SP\ C^o$ with target B^n are defined as:

$$D_S(C^o, B^n) \triangleq \frac{\sum_{j=1}^J d_s(c_{Xj}^o, Poly_{B^n})^2}{\{A_{B^n} + A_{C^o}\}} \quad (3)$$

$d_s(c_{Xj}^o, Poly_{B^n}) \triangleq$ Shortest distance of c_{Xj}^o from polygon $Poly_{B^n}$.

The sum of area term in the denominator is to normalize the match measure with respect to area of the patterns.

$$D_X(C^o, B^n) \triangleq \frac{\sum_{j=1}^J d_x(c_{Xj}^o, Poly_{B^n})^2}{\{A_{B^n} + A_{C^o}\} \times F} \quad (4)$$

$$d_x \triangleq \begin{cases} 0 & \text{If } c_{Xj}^o \text{ is within polygon } Poly_{B^n} \text{ otherwise} \\ \text{shortest distance of } c_{Xj}^o \text{ from the polygon } Poly_{B^n}. \end{cases}$$

$$F \triangleq \begin{cases} 0 & \text{If } A_{B^n} \cap A_{C^o} = 0. \\ 1 & \text{If } A_{B^n} \cap A_{C^o} > 0. \end{cases}$$

The computation of d_s and d_x is explained in Figure 1. D_S is a simple shape matching measure mentioned here for the purpose of comparing the matching results with the new match measure D_X . Some of the properties of D_X are:

1. Non-negative: $D_X(C^o, B^n) \geq 0$.
2. Non-symmetric: $D_X(C^o, B^n) \neq D_X(B^n, C^o)$.
3. If $D_X(A, B) = 0$ & $D_X(B, C) = 0$ But $D_X(A, C) = 0$ & $D_X(B, C) = 0$
 $\Rightarrow D_X(A, C) = 0$.
4. $\neq \Rightarrow D_X(A, B) = 0$.

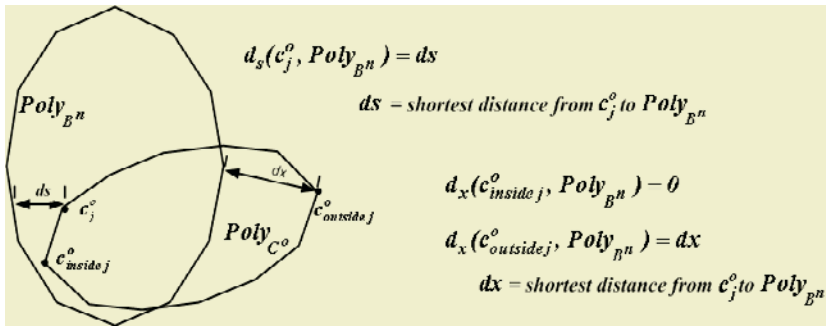


Fig. 1. This figure explains the computation of distance d_x and d_s of a control point on $Poly_{C^o}$ with $Poly_{B^n}$.

A SP, C^o is a match with target B^n when the match measure $D_X(C^o, B^n)$ is equal to zero. This happens when C^o is spatially coincident with target B^n or C^o lies entirely within B^n . However, in practice a C^o would be considered a match with B^n when $D_X(C^o, B^n)$ is smaller than a threshold, which is not critical. The presence of F term in the denominator of (4) ensures that the two contours of C^o and B^n match only when there is overlap between them.

The match measure D_H is defined as

$$D_H(C^o, B^n) \triangleq \sum_{i=1}^I |c_{Hi}^o - b_{Hi}^n| \quad (5)$$

for matching two patterns based on intensity and color information. Each bin Hi has three sub-bins corresponding to Y, C_r , and C_b channels and $|c_{Hi}^o - b_{Hi}^n|$ is the sum of the absolute differences of the sub-bins in bin Hi .

3.2 Pattern Matching

Here we consider matching targets with their SP when there is merging and splitting. Let the set of targets being tracked be represented by $B^1, B^2, \dots, B^n, \dots, B^N$. A frame consist of several SP s, which are $C^1, C^2, \dots, C^o, \dots, C^O$. A SP C^o can be from: 1. single target, 2. multiple targets merging together, 3. part of a target, which has split into multiple SP s, and 4. part of a target which has simultaneously merged with other targets and undergone split to give the SP . We focus on solving data association in cases 2 and 3 optimally and 4 is solved sub-optimally.

The merging of two or more targets is expressed with operator \oplus , i.e. $B^1 \oplus B^2 \oplus B^3 \oplus B^4$ denotes the merging of the 4 targets B^1, B^2, B^3 , and B^4 . The synthesized-pattern \bar{B} formed by merging targets B^1, B^2, B^3 , and B^4 will have a new convex hull. This convex hull of \bar{B} is obtained from the points on the convex hull of B^1, B^2, B^3 , and B^4 . Given N targets, the total number of different ways in which a new synthesized-pattern can be formed is $2^N - 1$. For example, for $N = 4$ the different possibilities for \bar{B} are $\{B^1, B^2, B^3, B^4, B^1 \oplus B^2, B^1 \oplus B^3, B^1 \oplus B^4, B^2 \oplus B^3, B^2 \oplus B^4, B^3 \oplus B^4, B^1 \oplus B^2 \oplus B^3, B^1 \oplus B^2 \oplus B^4, B^1 \oplus B^3 \oplus B^4, B^2 \oplus B^3 \oplus B^4, B^1 \oplus B^2 \oplus B^3 \oplus B^4\}$. Notationally any possible synthesized-pattern \bar{B} formed from merges can be written as

$$\bar{B} = B^{n(1)} \oplus B^{n(2)} \oplus \dots \oplus B^{n(p)} \oplus \dots \oplus B^{n(P)}. \quad (6)$$

Where P of the available targets have merged and $n(p)$ denotes the index of the targets used in synthesis of \bar{B} . For each \bar{B} the match measure $D_S(\bar{B}, C^o)$ can be computed to find the best match such that

$$\hat{P}, \hat{n}(p) = \arg \min_{P, n(1), \dots, n(p)} [D_S(\bar{B}, C^o)]. \quad (7)$$

This formulation considers all the possible merges of targets and the one which gives minimum match measure is finally chosen, thus giving an optimal solution. However, the minimization problem expressed in (7) is computationally prohibitive to solve in real time. The order of computation to find optimal match when targets merge is $O((2^N - 1) \times O)$. For $N = 12$ to 17 this would be quite a large number.

The optimal solution to the problem of splitting can be addressed by merging the SP s, C^o in all possible combinations and computing their match measure

with the different targets. The new synthesized pattern formed by merging different SP s is $\bar{C} = C^{o(1)} \oplus C^{o(2)} \oplus \dots \oplus C^{o(p)} \oplus \dots \oplus C^{o(P)}$. The problem is to compute all possible \bar{C} by changing P , the number of SP and $o(p)$ the indexes of SP . Thus the problem can be optimally solved as in the case of merges with a computational complexity of $O((2^O - 1) \times N)$. Some times the total number of SP can be quite large due to the presence of clutter, which would make 2^O a very large number. The total complexity to handle both splits and merges is $O((2^N - 1) \times O) + O((2^O - 1) \times N)$. Next we show how by using the new match measure D_X and dynamic programming the same problem can be solved in $O(N \times O)$.

3.3 Dynamic Programming Strategy for Efficient Pattern Matching

Dynamic Programming (DP) is a powerful nonlinear optimization technique, and is used here to solve the pattern matching problem by optimizing a function which evaluates the match between targets and SP s. The use of DP in solving a problem requires that the problem be divided into sub-problems and optimal solution of these sub-problems can be combined together to obtain optimal solution for the main problem. The properties of the distance function D_X facilitates the use of the DP strategy. In Figure 2 four targets $B^{n(1)}, B^{n(2)}, B^{n(3)}$, and $B^{n(4)}$ at time instant k are being tracked with a Kalman filter based tracker. The predictions for their shape and position are available for the next frame $k + 1$ and are denoted as, $\hat{B}^{n(1)}, \hat{B}^{n(2)}, \hat{B}^{n(3)}$, and $\hat{B}^{n(4)}$. These targets merge to give

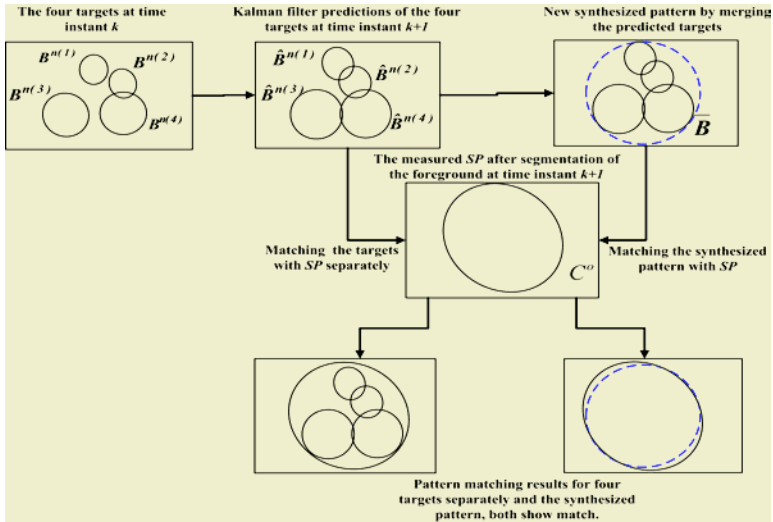


Fig. 2. This figure explains the pattern matching principle which enables use of dynamic programming strategy to speed up computations. A synthesized pattern $\bar{B} = \hat{B}^{n(1)} \oplus \hat{B}^{n(2)} \oplus \hat{B}^{n(3)} \oplus \hat{B}^{n(4)}$ is a match with $SP C^{so}$ by the distance measure D_X when $\hat{B}^{n(1)}, \hat{B}^{n(2)}, \hat{B}^{n(3)}$, and $\hat{B}^{n(4)}$ all match with C^{so} separately.

rise to a new synthesized pattern \overline{B} in frame $k + 1$. If a SP C^o is due to the merger of these four targets in frame $k + 1$ then \overline{B} will coincide with C^o and hence $D_X(\overline{B}, C^o)$ will be equal to zero. Now if D_X of $\hat{B}^{n(1)}, \hat{B}^{n(2)}, \hat{B}^{n(3)}$, and $\hat{B}^{n(4)}$ is computed with C^o separately then each one of them will be equal to zero. Because all the control points of these targets lie within the polygon formed by the control points of \overline{B} which is a match with C^o . Therefore the problem of pattern matching when targets undergo merge to form a new synthesized pattern \overline{B} is sub-divided to the problem of finding all targets which match SP , C^o separately by the match measure D_X . If the predictions of all the targets $B^{n(1)}, B^{n(2)}, \dots, B^{n(P)}$ match with C^o separately then \overline{B} formed by merging these targets is optimal match for C^o .

When the targets undergo splitting to give rise to more SP s then the targets in the scene then the same DP strategy as above can be applied by reversing the roles of B^n and C^o .

4 Kalman Filter Based Tracking

The system uses two Kalman filters to track each target. Theoretically our system can be classified as a multiple model system [15]. The motion and position of a target is tracked by tracking the centroid of the ellipse modelling the target. Assuming that video rate sampling of 25 frame/sec is fast enough we model the motion of the targets with a constant velocity model. The shape of the target is tracked by tracking the J control points representing the shape of the target. The motion of these control points are approximated by affine motion model, which has been widely used in computer vision for segmentation and tracking [16][17]. The change of shape of the targets as they move away or towards the camera is accounted by the parameter b_s^n of the n^{th} target. The details of the Kalman filter equations and their derivation can be obtained from [18].

From the discussion in Section 3 it can be said that there are three types of matches possible. Each of these and their different methods for updating the filter parameters are:

1. The targets which have not undergone merge or splits, match their corresponding SP with match measures D_H and D_X . The motion, position, and shape attributes of these targets are updated by the Kalman Filter *estimates*.
2. For matches where a target has split into multiple SP s the shape feature of the target is updated by Kalman Filter *predictions* but position and motion are updated by Kalman Filter *estimates*. In the latter case the new measurement of the centroid is the mean of all centroids of the different SP s, which match the target.
3. The shape, position and motion features of targets, which have merged or have undergone simultaneous merges and splits are updated by their Kalman Filter *predictions* for all attributes motion, position, and shape.

The result of tracking as a co-operative effort between pattern matching and Kalman filter based tracking are shown on a test image sequence in Figure 3.

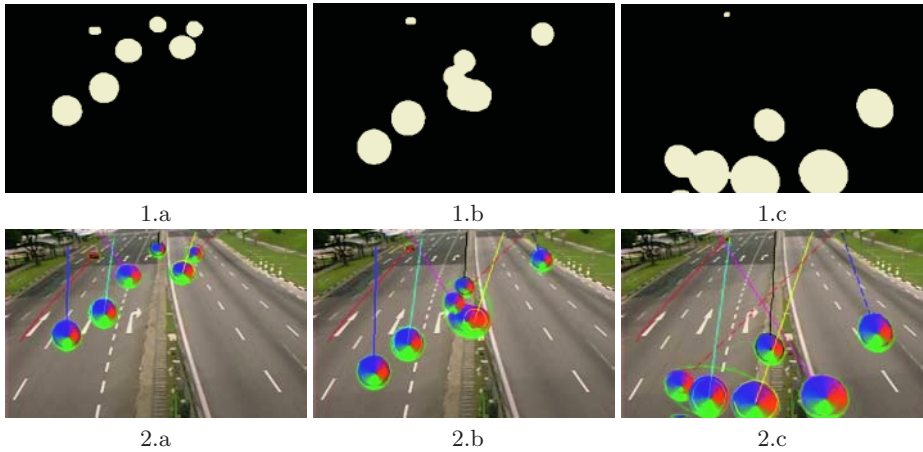


Fig. 3. Images 1.a,b,c, show the segmentation results for eight targets in the FOV and images 2.a,b,c show the tracking results. Images 1.b and 2.b show a case where four targets were merged into one *SP*. Here each of the eight targets were properly tracked even as they underwent multiple merges and splits. Please note here that all the targets are similarly colored so a correspondence based tracking is likely to fail.

These images show the algorithm's ability to handle multiple merges. The video was made by overlaying artificial targets on a real video. In spite of many instances of merges the targets position and shape have been quite accurately tracked.

5 Track Initialization

Accurate initialization of the position, motion, and shape parameters of a target is an important step, which must be accomplished in the presence of clutter. There are two types of track initializations that needs to be handled: 1. the initial bootstrapping and 2. when the tracking is in progress.

Attributed graphs are very useful for initializing Multi-Target Tracking (MTT) systems as it provides a technique for incorporating both spatial and temporal information of the targets in decision making. Graphs for target track initialization and tracking have been used in [19] and [8], respectively. Our system initializes a new target for tracking only when a target's reliable measurements are available in the past τ frames. This property makes the initialization accurate and the tracker stable.

Automatic initialization of target tracks is done by using an attributed graph of the *SPs* in τ frames, as shown in Figure 4. The attributes of each node in the graph are: 1. the frame number, 2. the centroid, 4. shape parameters, 5. color histogram of the *SP*, 6. parent id and child id. Edges are present between nodes whose frame number differ by 1 as shown in Figure 4. The weights of these

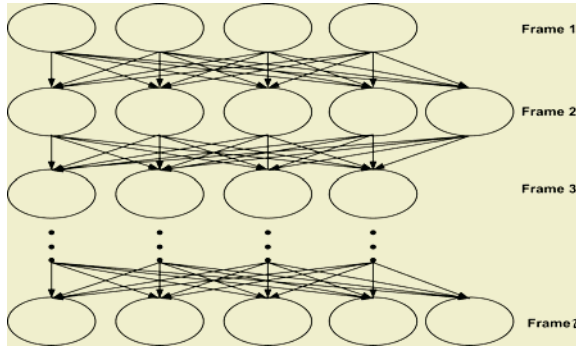


Fig. 4. This figure shows the structure of the attributed graph used for initialization of tracking.

edges is the weighted sum of match measures D_H and D_S between the nodes. The nodes with frame number 1 are considered as source nodes and nodes with frame number τ are considered as destination nodes. For all source nodes the shortest path to all destination nodes are computed using Dijkstra's algorithm. Amongst these shortest paths different paths have different sum-of-weights. Of these paths the path with smallest sum-of-weights is chosen and is called the path of least sum-of-weights. This path is considered a valid target track. The nodes of this path is removed from the graph giving rise to a new graph with reduced number of nodes and edges. The same process is repeated for the new graph until there is no node from any one of the τ frames in the graph or the path of least sum-of-weights amongst the computed shortest paths at any iteration is greater than a heuristic threshold.

Another problem is initialization of tracking for new targets, which enter the FOV or appear in the FOV due to resolution of occlusion, when tracking of other targets are in progress. To solve this problem an attributed graph of SPs which have no match with the targets being tracked is maintained. The path of least sum-of-weights amongst all the shortest paths from the source nodes to the destination nodes is computed as described earlier. The source nodes are from the first layer formed by the unmatched SPs in frame $(k - \tau + 1)$, where k is the current frame number. The destination nodes are the unmatched SPs of frame k . A new target is confirmed by appearance of least sum-of-weights path amongst all the shortest path possible from the source nodes to the destination nodes. All the nodes of this path are removed from the attributed graph.

6 Bayesian Network Based Classifier

Bayesian network classifiers provide a probabilistic framework, which allow the power of statistical inference and learning to be combined with the temporal and contextual knowledge of the problem [20]. We have used Bayesian network for classification of targets in image sequences from a stationary camera. The

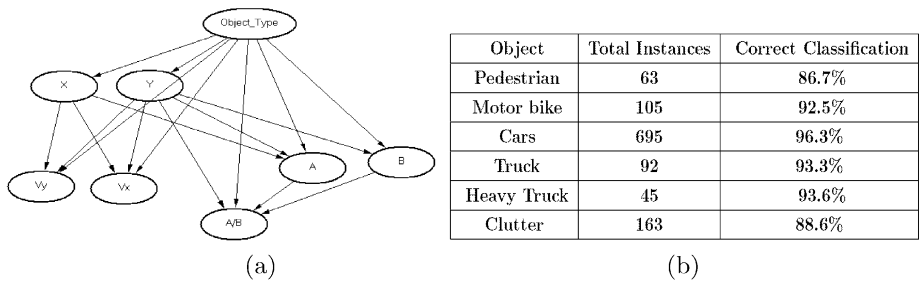


Fig. 5. (a) Bayesian Network structure used for target classification and (b) the classification results obtained from this classifier.

different classes of targets considered are pedestrians, motorcycles, cars/vans, trucks/buses, heavy trucks, and clutter. In general it is difficult to model a deterministic relationship between the size, shape, position, and motion features to the object class due to perspective effects. For example a car close to the camera may be of the same size as a truck far from the camera; similarly a pedestrian passing close to the camera may show motion in image space which is similar to the motion of a fast moving car far from the camera. Furthermore there are internal dependencies amongst the features. For example, the speed and size of an object is dependent upon its position. Thus, to establish a relationship between the various image features of a target and its type, and to model the conditional dependencies amongst the features we use a Bayesian Network based classifier. The use of motion and position parameters of a target from tracking module makes the classification more robust. For example the size and shape of a moving motorcycle and a pedestrian may be similar but their motion and position are usually different.

Figure 5(a) shows the proposed Bayesian Network model. Each node is a variable and the object node is the root node. The seven measurement nodes are X , Y (the x, y co-ordinates of the target in image space), V_x , V_y (the x, y components of the target velocity in image space), A , B , the major and minor axis of the ellipse modelling the target's shape and A/B the aspect ratio of the ellipse. Figure 5(b) shows the classification results from several image sequences. In each of these case the ground truth was manually obtained.

7 Camera Calibration and 3D Tracking

To convert the tracking results in image co-ordinate space to world co-ordinate space we need to know the perspective transformation matrix P . We use the technique similar to that of [21] to compute P . The XY plane of the world co-ordinate system is aligned with the ground plane of the scene and the Z axis is perpendicular to the ground plane. The image co-ordinates are related to the world co-ordinate as follows:

$$\begin{bmatrix} x_i \\ y_i \\ \lambda \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (8)$$

From (8) it can be easily shown that if the height of a point Z_w in the world co-ordinates is known along with its image co-ordinates, then the corresponding world co-ordinate location X_w , Y_w of the point can be obtained as:

$$X_w = \frac{(p_{12} - p_{32}x_i)(p_{23} - p_{33}y_i) + (p_{13} - p_{33}y_i)(p_{32}y_i - p_{22})}{(p_{32}y_i - p_{22})(p_{31}x_i - p_{11}) - (p_{12} - p_{32}x_i)(p_{21} - p_{31}y_i)} Z_w \quad (9)$$

$$Y_w = \frac{(p_{12} - p_{32}x_i)(p_{24} - p_{34}y_i) + (p_{24} - p_{34}y_i)(p_{32}y_i - p_{22})}{(p_{32}y_i - p_{22})(p_{31}x_i - p_{11}) - (p_{12} - p_{32}x_i)(p_{21} - p_{31}y_i)} + \frac{(p_{21} - p_{31}y_i)X_w}{(p_{32}y_i - p_{22})} + \frac{(p_{23} - p_{33}y_i)Z_w}{(p_{32}y_i - p_{22})} + \frac{(p_{24} - p_{34}y_i)}{(p_{32}y_i - p_{22})} \quad (10)$$

After the targets are classified, a model height of targets in different categories as shown in Table 1 is used to estimate the 3D position and motion of the targets. The model heights are a rough guides to the height of a 3D point on top of the target. In simulations it was found that an error of ± 0.5 meters in the height estimate translates to about $\pm 10\%$ error in speed estimate. To get an accurate estimate of the world speed of a target, a point which is on top of the target is selected using heuristics based on the camera view.

Table 1. Model height values for the different classes of targets obtained by averaging the different heights of objects in a class.

Pedestrian	Motorcycles	Cars/Vans	Trucks/Buses	Heavy Truck/Double-Decker
1.7m	1.5m	1.7m	3.0m	4.0m

8 Results

We show the results of tracking for both articulated and non-articulated objects. In all the results the green ellipses are used to show the measurements obtained in every frame. The targets and their tracks are shown with same color, which is different for different targets as far as possible. In Figure 6 we show the robust tracking results for a real traffic scene and one of the image sequence of PETS data set. Complete tracking video for traffic scene can be seen in ‘video.avi’ in the supplementary files. Here simultaneous tracking of upto 17 targets can be seen. There are instances here when the targets were completely occluded, a target splitted in to more than one *SPs* and some targets merged to give one *SP* and there were lot of clutter. In all these cases tracking continued without errors and there was no wrong initialization.

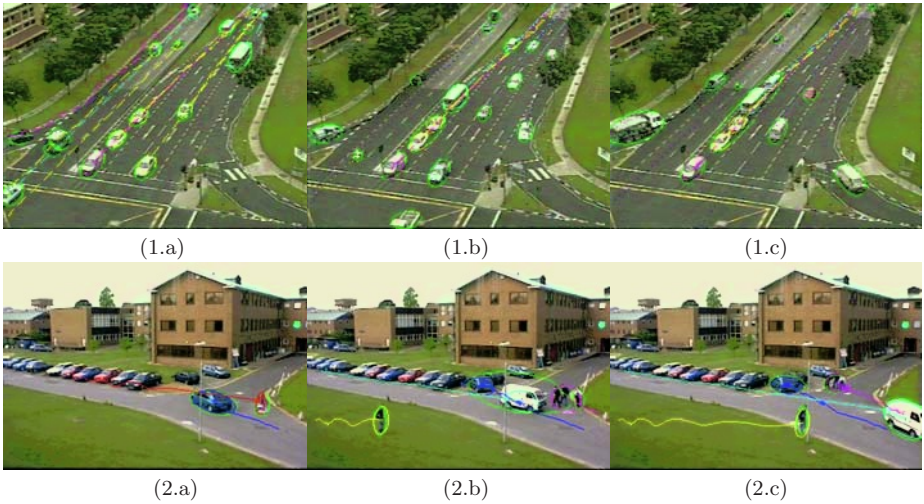


Fig. 6. Images 1.a,b,c show the tracking results for a traffic scene. (a) shows a case where the 2nd target from bottom left split into two measurements, (b) shows a frame where 17 targets are being tracked simultaneously, and (c) shows a case where four vehicles merged into one measurement. Images 2. a,b,c shows the tracking results on an image sequence from PETS2001 data set. Image 2.b shows a case where an instance of simultaneous merging and splitting has been handled properly.

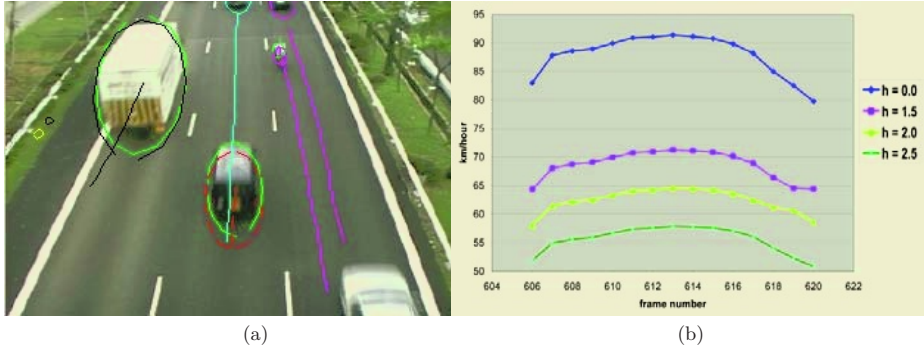


Fig. 7. Image (a) shows 2D tracking results of a frame in a test image sequence. The vans in the center of image (a) was moving with a constant speed of 65 km/hour as read from its speedometer. Plot (b) shows the computed speed of the vehicle for different height estimates denoted by the parameter 'h' and expressed in meters.

Figure 7 shows the results of 3D tracking algorithm proposed in the paper. Figure 7(a) shows the 2D tracking results and 7(b) shows the plot of the computed speed of the black van, being tracked in the center of the image 7(a), at different estimates of height. When the estimated height of the vehicle is taken to be zero then there is significant error in the speed estimates. The speed esti-

mates are in the range of 80-92 km/hour when the actual speed is 65km/hour as read from the speedometer of the vehicle. The speed estimates for other values of height, such as 1.5 meters, 2 meters, and 2.5 meters are close to the actual speed of 65km/hour. The actual height of the van is 2 meters. This accurate estimation of the speed of the target allows for detecting a vehicle's acceleration as well.

9 Conclusions

We have addressed several problems for robust and reliable tracking and classification of multiple targets in image sequences from a stationary camera. A new efficient algorithm based on DP strategy for pattern matching was proposed, which can handle data association during complex splitting and merging of the targets. When this technique is combined with Kalman filter based tracking, it is possible to preserve the labels of the targets even when they cross each other, or get completely or partially occluded by background or foreground objects. An attributed graph based technique was proposed to initialize the tracks. Using a Bayesian network based classification and a simple camera calibration we have obtained accurate 3D tracking results for vehicles. Results have been shown where the tracker can handle up to 17 targets simultaneously. At present this system is being used to detect potential accident behavior between pedestrians and vehicles in traffic videos.

References

1. Y. Bar-Shalom, "Tracking methods in a multitarget environment," *IEEE Transactions on Automatic Control*, vol. Vol. AC-23, No. 4, pp. 618–626, August 1978.
2. S. Blackman, *Multiple-Target Tracking with Radar Application*. Artech House, 1986.
3. N. Paragios and R. Deriche, "Geodesic active regions for motion estimation and tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 266–280, March 2000.
4. S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, "Tracking groups of people," *Computer Vision and Image Understanding*, vol. 80, pp. 42–56, 2000.
5. O. Javed and M. Shah, "Tracking and object classification for automated surveillance," in *European conference on Computer Vision*, 2002.
6. I. Haritaoglu, D. Harwood, and L. Davis, "W4: Who, when, where, what: A real time system for detecting and tracking people," in *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition (FG'98)*, pp. 222–227, April 1998.
7. I. Haritaoglu, D. Harwood, and L. Davis, "W4s: A real time system for detecting and tracking people in 2.5d," *Fifth European Conference on Computer Vision*, vol. June, pp. 877–892, 1998.
8. G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, "Event detection and analysis from video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 873–889, August 2001.

9. D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. Vol. AC-24, No. 6, pp. 843–854, December 1979.
10. I. Cox and M. Miller, "On finding rank assignments with application to multitarget tracking and motion correspondence," *Aerosys*, vol. Vol. 32, pp. 486–489, Jan. 1996.
11. H. Tao, H. S. Sawhney, and R. Kumar, "Object tracking with bayesian estimation of dynamic layer representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24 No. 1, pp. 75–89, 2002.
12. P. Kumar, S. Ranganath, and W. Huang, "Queue based fast background modelling and fast hysteresis thresholding for better foreground segmentation," in *Proceedings of The 2003 Joint Conference of the Fourth ICICS and PCM*, (Singapore), p. 2A2.5, December 2003.
13. L. Li, W. Huang, Y. G. Irene, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proceedings of ACM Multimedia*, pp. 2–10, November 2003.
14. A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 476–480, May 1999.
15. E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey.," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 4, pp. 103–123, 1998.
16. R. Cipolla and A. Blake, "Surface orientation and time to contact from image divergence and deformation," in *Proceedings of Second European Conference on Computer Vision* (G. Sandini, ed.), (S. Margherita, Ligure, Italy), pp. 187–202, Springer-Verlag, Berlin, Heidelberg, New York, May 1992.
17. Q. Zheng and R. Chellappa, "Automatic feature point extraction and tracking in image sequences for unknown camera motion," in *Proceedings of International Conference on Computer Vision*, (Berlin, Germany), pp. 335–339, May 1993.
18. P. Kumar, "Multi-body tracking and behavior analysis." Ph.D. Thesis, National University of Singapore.
19. J. K. Wolf, A. M. Viterbi, and G. S. Dixon, "Finding the best set of k-paths through a trellis with application to multitarget tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 287–295, 1989.
20. A. Mittal and C. L. Fah, "Characterizing content using perceptual level features and context cues through dynamic bayesian framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. Special Issue on Conceptual and Dynamical Aspects of Multimedia Content Description., under submission.
21. K. Matsui, M. Iwase, M. Agata, T. Tanaka, and N. Onishi, "Soccer image sequence computed by a virtual camera," in *Proc. Conf. on Computer Vision and Pattern Recognition*, pp. 860–865, 1998.

A Linguistic Feature Vector for the Visual Interpretation of Sign Language

Richard Bowden^{1,2}, David Windridge¹, Timor Kadir²,
Andrew Zisserman², and Michael Brady²

¹ CVSSP, School of EPS,
University of Surrey, Guildford,
Surrey, UK

{r.bowden,d.windridge}@eim.surrey.ac.uk

² Department of Engineering Science,
University of Oxford,
Oxford, UK.

{timork,az,jmb}@robots.ox.ac.uk

Abstract. This paper presents a novel approach to sign language recognition that provides extremely high classification rates on minimal training data. Key to this approach is a 2 stage classification procedure where an initial classification stage extracts a high level description of hand shape and motion. This high level description is based upon sign linguistics and describes actions at a conceptual level easily understood by humans. Moreover, such a description broadly generalises temporal activities naturally overcoming variability of people and environments. A second stage of classification is then used to model the temporal transitions of individual signs using a classifier bank of Markov chains combined with Independent Component Analysis. We demonstrate classification rates as high as 97.67% for a lexicon of 43 words using only single instance training outperforming previous approaches where thousands of training examples are required.

1 Introduction

Sign Language is a visual language and consists of 3 major components: finger-spelling – used to spell words letter by letter; word level sign vocabulary – used for the majority of communication; and non manual features – facial expressions and tongue, mouth and body position.

Within the literature there has been extensive work performed on finger-spelling, e.g. [1,7]; but this area is a small subset of the overall problem. For word level sign recognition, the most successful methods to date have used devices such as data-gloves and electromagnetic/optical tracking, rather than monocular image sequences, and have achieved lexical sizes as high as 250 signs or words [3,4,5,6]. However, without using such devices recognition is typically limited to around 50 words and even this has required a heavily constrained artificial grammar on the structure of the sentences [8,11].

Our objective is large lexicon (word level) sign recognition from monocular image sequences. Traditionally, sign recognition has been based upon extensive training. However, this limits scalability as the acquisition of labelled training data is expensive and time consuming. Thus our aim is a method with low training requirements. Additionally, we aim for transferable learning, so that signs learnt from one individual enable signs from another individual to be recognized. In a perfect world ‘one-shot-training’ should be capable of addressing both of these aims, and this work presents an approach capable of achieving high recognition rates across individuals with as little as a single training instance per sign/word.

Previous approaches to word level sign recognition borrow from the area of speech recognition and rely heavily upon tools such as Hidden Markov Models (HMMs) [8,11,12] to represent temporal transitions. In turn this has meant extensive training sets have been required, for example Vogler and Metaxas [12] require 1292 training examples for a 22-word lexicon.

The novelty of the work presented here is that we structure the classification model around a linguistic definition of signed words, rather than a HMM. This enables signs to be learnt reliably from just a handful of training examples, and we have been able to reach the state of the art (49 words) using just this training set.

The classification process is divided into two stages. The first generates a description of hand shape and movement at the level of ‘the hand has shape 5 (an open hand) and is over the left shoulder moving right’. This level of feature is based directly upon those used within sign linguistics to document signs. Its broad description aids in generalisation and therefore significantly reduces the requirements of further stages of classification. In the second stage, we apply Independent Component Analysis (ICA) to separate the channels of information from uncorrelated noise. Final classification uses a bank of Markov models to recognise the temporal transitions of individual words/signs.

In the following section, we describe the system architecture and its motivation. Section 3 then presents the vision elements used to track the user; in Section 4, the results of tracking are classified in terms of a linguistically inspired description of activity. Section 5 describes the second stage of classification, in which the temporal aspects of signs are both learnt and recognised. Finally, Section 6 presents results and a discussion of the techniques used and Section 7 highlights our future work.

2 Overview

A graphical overview of the system is given in Figure 1. Our approach is based upon a novel two stage classification:

Classification stage I: Raw image sequences are segmented in order to extract the shapes and trajectories of the hands in the monocular image sequence. The initial classification stage converts these into a “viseme” representation (the visual equivalent of a phoneme) taken from sign linguistics [2]:

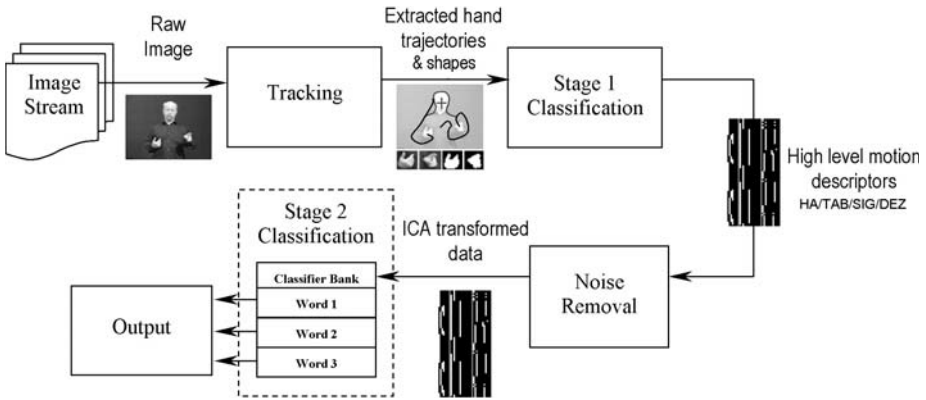


Fig. 1. Block diagram showing a high level overview of the stages of classification.

- HA** Position of the hands relative to each other
- TAB** Position of hands relative to key body locations
- SIG** Relative movement of the hands
- DEZ** The shape of the hand(s)

This HA/TAB/SIG/DEZ notation provides a high-level feature descriptor that broadly specifies events in terms such as *hands move apart*, *hands touch* or *right hand on left shoulder*. This description of scene content naturally generalises temporal events, hence reduces training requirements. This is described in more detail in Section 4.

Classification stage II: Each sign is modelled as a 1st order Markov chain in which each state in the chain represents a particular set of feature vectors (denoted *symbols* below) from the stage I classification. The Markov chain encodes temporal transitions of the signer’s hands. During classification, the chain which produces the highest probability of describing the observation sequence is deemed to be the recognised word. In the training stage, these Markov chains may be learnt from a single training example.

Robust Symbol Selection: An appropriate mapping from stage I feature vectors to symbols (representing the states in the Markov chains) must be selected. If signs were produced by signers without any variability, or if the stage I classification was perfect, then (aside from any computational concerns) one could simply use a one-to-one mapping; that is, each unique feature vector that occurs in the course of a sign is assigned a corresponding state in the chain. However, the HA/TAB/SIG/DEZ representation we employ is binary and signers do not exhibit perfect repeatability. Minor variations over sign instances appear as perturbations in the feature vector degrading classification performance.

For example the BSL sign for ‘Television’, ‘Computer’ or ‘Picture’ all involve an iconic drawing of a square with both hands in front of the signer. The hands move apart (for the top of the square) and then down (for the side) etc. Ideally, a HMM could be learnt to represent the appropriate sequence of

HA/TAB/SIG/DEZ representations for these motions. However the exact position/size of the square and velocity of the hands vary between individual signers as does the context in which they are using the sign. This results in subtle variations in any feature vector however successfully it attempts to generalise the motion.

To achieve an optimal feature-to-symbol mapping we apply Independent Component Analysis (ICA). Termed feature selection, this takes advantage of the separation of correlated features and noise in an ICA transformed space and removes those dimensions that correspond to noise.

3 Visual Tracking

For completeness, we briefly describe the head and hand tracking, though we do not consider it to be the innovative part of our work.

Signers generally face the viewer as directly as possible to ease understanding and remove ambiguities and occlusions that occur at more oblique angles. The system uses a probabilistic labelling of skin to roughly locate the face of a signer. This, coupled with a contour model of the head and shoulders, provides a body-centred co-ordinate system in which to describe the position and motion of the hands. The 2D contour is a coarse approximation to the shape of the shoulders and head and consists of 18 connected points as shown in Figure 2a. The contour is a mathematical mean shape taken from a number of sample images of signers. The contour is then fitted to the image by estimating the similarity transform which minimises the contour's distance to local image features.

Estimates for key body locations, as indicated in Figure 2a, are placed relative to the location of the head contour. This means that as the contour is

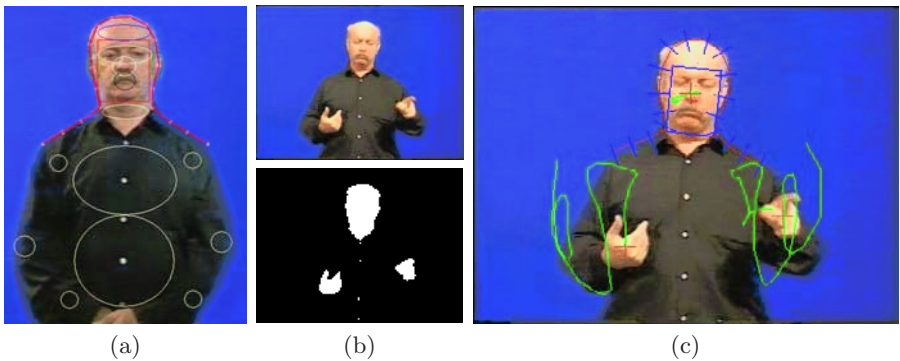


Fig. 2. (a) The 2D contour of the head and shoulders fitted to an image with positions and approximate variances for key body locations. (b) Tracking the hands using colour: top — an original frame from the signing video sequence; bottom — the binary skin map (white pixels denote a Mahalanobis distance < 3). (c) The result of contour tracking and trajectory of the hands over time.

transformed to fit the location of the user within the video stream, so the approximate locations of the key body components are also transformed. Figure 2c shows the system tracking the upper torso and hands of an active signer, the trails from the hands show the path taken over the last 50 frames.

4 Stage I Classification

The 5 HA, 13 TAB, 10 SIG and 6 DEZ states we currently use are listed in Table 1 and are computed as follows:

HA: the positions of the hands relative to each other is derived directly from deterministic rules on the relative x and y co-ordinates of the centroids of the hands and their approximate area (in pixels).

TAB: the position of the hands is categorised in terms of their proximity to key body locations (shown in Figure 2) using the Mahalanobis distance computed from the approximate variance of those body parts.

SIG: the movement of the hands is determined using the approximate size of the hand as a threshold to discard ambient movement and noise. The motion is then coarsely quantised into the 10 categories listed in Table 1.

DEZ: British Sign Language has 57 unique hand-shapes (excluding finger-spelling) which may be further organised into 22 main groups [2]. A visual exemplar approach is used to classify the hand shape into six (of the 22) groups. This is described in detail below.

Table 1. The high level features after stage I classification.

HA	TAB	SIG	DEZ
1. Right hand high	1. The neutral space	1. Hand makes no movement	1. 5
2. Left hand high	2. Face	2. Hand moves up	2. A
3. Hands side by side	3. Left Side of face	3. Hand moves down	3. B
4. Hands are in contact	4. Right Side of face	6. Hand moves left	4. G
5. Hands are crossed	5. Chin	7. Hand moves right	5. H
	6. R Shoulder	8. Hands moves apart	6. V
	7. L Shoulder	9. Hands move together	
	8. Chest	10. Hands move in unison	
	9. Stomach		
	10. Right Hip		
	11. Left Hip		
	12. Right elbow		
	13. Left elbow		

Figure 3b shows the features generated by the system over time. The horizontal binary vector shows HA, SIG, TAB and DEZ in that order delineated by grey bands. The consistency in features produced can clearly be seen between examples of the same word. It is also possible to decode the vectors back into a textual description of the sign in the same way one would with a dictionary. The feature vector naturally generalises the motion without loss in descriptive ability.

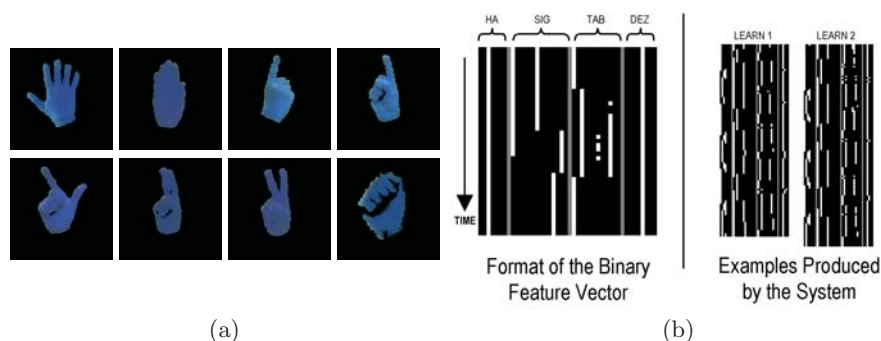


Fig. 3. (a) Examples of hand-shapes used in British Sign Language – from top-left clockwise ‘5’, ‘B’, ‘G’, ‘G’, ‘A’, ‘V’, ‘H’, ‘G’. (b) Graphical representation of feature vectors for different occurrences of signs demonstrating the consistency of feature description produced and two instances of the sign ‘learn’.

Linguistic evidence [10] suggests that sign recognition is based primarily on the dominant hand (which conveys the majority of information). For this reason, we currently discard the non dominant hand and concatenate the HA, TAB, SIG and DEZ features to produce a 34 dimensional binary vector which describes the shape and motion in a single frame of video.

4.1 Hand Shape Classification

Within the lexicon of 49 words used to date in this work, six main sign groups appear, denoted ‘5’, ‘A’, ‘B’, ‘G’, ‘H’ and ‘V’ following the definition in [2]. Typical examples of these are shown in Figure 3. Our objective is to classify hand-shapes into one of these six groups.

The visual appearance of a hand is a function of several factors which a hand shape classifier must take into account. These include: pose, lighting, occlusions and intra/inter-signer variations. To deal with such variations, we adopt an exemplar based approach, where many visual exemplars correspond to the same sign group.

Segmentations of the hands are first obtained from the tracking stage discussed in Section 3. Hand-shape is represented as a binary mask corresponding to the silhouette of the hand and these masks are normalised for scale and orientation using their first and second moments. Learning proceeds by generating a set of normalised masks from training data (see Section 6) and clustering these to form an exemplar set. We use a greedy clusterer with a normalised correlation distance metric. A threshold on this distance controls the degree of grouping.

Novel hand-shapes are then classified by matching their normalised binary masks to the nearest one in the exemplar set using normalised correlation as a distance metric. Similar approaches have been used by several previous authors, for example [7,9].

This exemplar approach has a number of attractive properties. Different hand-shapes in the same group, and variations in appearance of the same hand-

shape due to different poses or signer variation, may be represented by separate exemplars assigned to the same group label.

While it is clear that this basic hand classifier cannot distinguish between all poses and shapes, we demonstrate that it complements the HA, TAB and SIG features, hence is *sufficient* to discriminate between a fixed lexicon of 49 words. For much larger lexicons the representation may be augmented in various ways, for example through the use of internal features to capture information about the positions of the fingers in closed hand-shapes, or by allowing discrimination within hand-shape groups. Set at an operating point chosen to give 1770 exemplars, the hand-shape classifier achieves an average correct classification rate of 75%. The results presented in Section 6 use this operating point.

5 Stage II Classification

5.1 Training

In order to represent the temporal transitions which are indicative of a sign, we make a 1st order assumption and construct a 1st order Markov chain for each word in the lexicon. However, this assumption requires that an ergodic model be constructed. With a 34 dimensional binary feature vector, this would result in a chain with $2^{28} \times 6$ states (5 HA + 13 TAB + 10 SIG multiplied by the 6 mutually exclusive DEZ features) and over 2.6×10^{17} possible transitions requiring a prohibitive amount of a storage. However, as can be seen in Figure 3b, the number of transitions in each word is typically small and governed by the duration of the sign and the capture rate (in this case 25Hz).

It is also evident that out of the $2^{28} \times 6$ possible states only a small subset ever occur and that there are even fewer transitions than combinatorially possible, due to the physical limitations of the human body. Therefore, we build only as much of the ergodic model as is needed. This is done by adding new feature states to the state transition matrix as they occur during training. The result is a sparse state transition matrix, $P_w(s_t|s_{t-1})$, for each word w giving a classification bank of Markov chains.

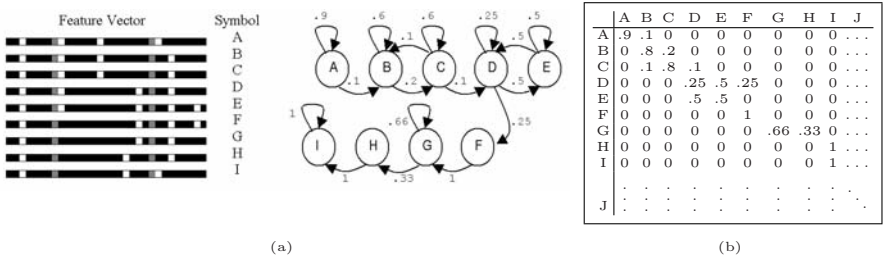


Fig. 4. The process of classifier chain construction ($P_w(s_t|s_{t-1})$): (a) first, a look up table is built mapping binary feature vectors to symbols and, (b) then a sparse ergodic state transition matrix representing the Markov chain can be generated.

5.2 Feature to Symbol Mapping

Mapping feature vectors directly onto states in the Markov chain as described above results in unsatisfactory classification performance. Minor variations across instances of the same sign and the rule-based stage I classification produces perturbations, or noise, in the binary feature vector resulting in misclassification.

One way to mitigate these effects is to use clustering to group ‘similar’ events together. In such a scheme, the mapping is no longer one-to-one; sets of stage I feature vectors map onto single symbols used for the Markov chain. However, in our binary feature space the concept of similarity, or distance, cannot be assumed to be Euclidean. Instead, we should *learn* this distance metric from the training data.

Our approach is to apply Independent Component Analysis (ICA) to the training data, the result of which is a transformation matrix, which, when applied, transforms the binary feature space into a space where an Euclidean metric can be used. Intuitively, the effect of the ICA is to move features that co-occur in the training data closer together and conversely those that are independent, further apart.

ICA attempts to separate the correlated features from uncorrelated noise; each incoming feature vector is transformed into the ICA space and an exhaustive feature selection process performed to decide which of the ICA transformed features are important to classification and which constitute noise. At each step a single feature, i.e. a single component of the transformed vector, is discarded. The feature to be removed is selected such that the overall performance in terms of classification of the reduced feature vector is maximised.

Once an ICA transformation matrix has been learnt, a look-up table (LUT) is generated from the training data to map the ICA transformed features to symbols for use in Markov chains.

5.3 Classification

During classification, the model bank is applied to incoming data in a fashion similar to HMMs. A sliding temporal window T is used to check each of the classifiers in turn against the incoming feature vectors. The objective is to calculate that chain which best describes the incoming data i.e. has the highest probability that it produced the observation sequence s . The probability of a model matching the observation sequence is calculated as $P(w|s) = \pi \prod_{t=1}^l P_w(s_t|s_{t-1})$, where $1 < l < T$ and $P_w(s_t|s_{t-1}) > 0, \forall t$, π is the prior probability of a chain starting in any one of its states. However, setting $\pi = 1$ and the stopping criteria $P_w(s_t|s_{t-1}) > 0, \forall t$, provides some robustness to the problems of sign co-articulation.

Of course for continuous recognition, this introduces a bias for short signs as $P(w|s)$ decreases as l increases. It is therefore necessary to use both the model match probability $P(w|s)$ and the number of observations the model describes, l , to find the sequence of words that best explain the observations over time. This

is done by maximising the overall probability using a Viterbi algorithm. The use of the Viterbi algorithm adds the problem that our approach has no model for silence. To overcome this, a running average of model match probability is used to synthesis a silence model within the Viterbi decoder. For isolated word recognition the chain which describes the largest consecutive segment of the observation sequence (l) can be used which overcomes the problems of word length bias.

6 Performance Results

The training and test data consists of 49 randomly selected words (listed in the appendix). Unlike previous approaches, signs were not selected to be visually distinct but merely to represent a suitable cross section of signs to allow a short conversation to take place.

Video sequences were collated for a single person performing the 49 signs. Each sign was repeated numerous times within the dataset resulting in a total of 249 individual signs, averaging 5 repetitions for each sign. The video was hand labelled for ground truth. A single instance of each sign was selected for training and the remaining 200 signs retained as an unseen test set. A classifier bank was learnt consisting of 49 Markov chains, one for each individual sign. The unseen data was then presented to the classifier bank and the most probable word determined using the approach described in Section 5. The output of the classifier bank was then compared to the ground truth to determine if a successful classification had been made. The results are presented in Figure 5.

Figure 5 shows the classification performance as a function of the number of features selected by ICA feature selection. The ICA transformed data results in a performance boost from 73% up to 84% percent classification rate beyond the 6 feature mark.

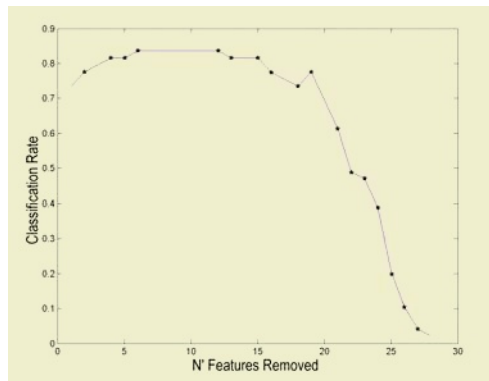


Fig. 5. Classification performance as a function of number of features removed in the ICA feature selection.

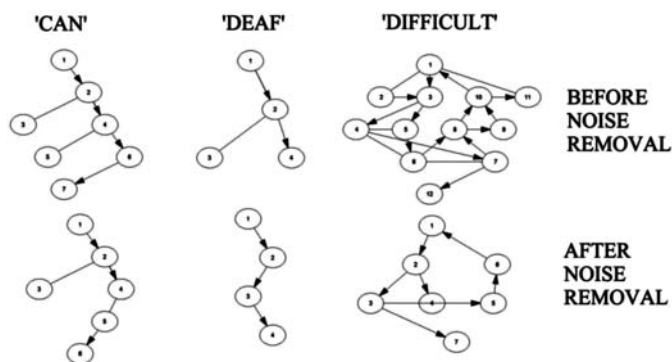


Fig. 6. Chains representing temporal transitions for both raw and ICA transformed motion descriptors for the words ‘can’, ‘deaf’ and ‘difficult’.

By selecting a lexical subset which does not contain ambiguous signs, for example by removing those signs for which facial gesture or contextual grammar are a significant cue, the results improve to 97.67% classification rate on a lexicon of 43 words. This compares well with previous approaches. It is important to note that these high classification results have been obtained without constraints on grammar and furthermore with only a single training instance per sign. This can be compared to other viseme level approaches based upon HMM’s where thousands of training examples are required to achieve similar levels of accuracy [8,11,12].

Figure 6 shows the results of ICA feature selection upon the transitions made by the feature vector. It can clearly be seen that the ICA transformed data produces more compact models. This is particularly evident for the word ‘difficult’ where the complex transitions between 12 states has been simplified through the removal of noise down to 7 states. More interesting to note is that the word ‘deaf’ is simplified to a left-right transition model almost identical to that which would be assumed for a HMM approach, however, many other signs do not fit this simplistic assumption. For example, the sign ‘difficult’ repeats 2 or more times as the thumb of the right hand taps against the left open palm. The resulting chain, shown on the right in Figure 6, is clearly cyclic with 2 possible end states. Such a chain intuitively fits with this type of repeating sign. It is not clear how such signs can be forced to map to a left-right transition model without some loss of information. This serves to illustrate that the assumptions on which speech recognition are based do not naturally extend to sign.

7 Conclusions

Our current demonstrator runs at 25fps on a 1GHz laptop and is capable of operating in an unconstrained indoor environment with background clutter. Due to the generalisation of features, and therefore the simplification in training,

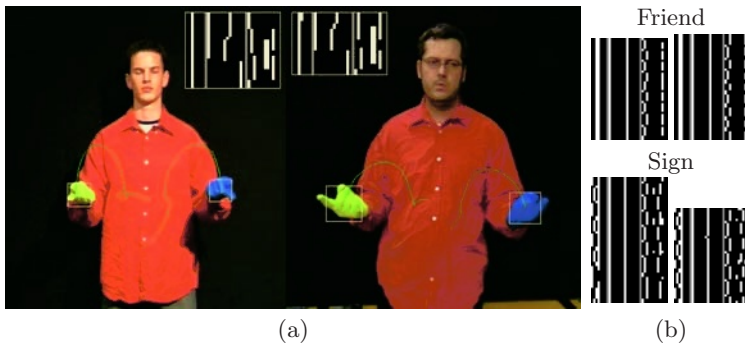


Fig. 7. Generalisation of feature vectors across different individuals: (a) two people signing the sign ‘different’; (b) binary feature vectors for two people for the signs ‘friend’ and ‘sign’. The signs are correctly classified.

chains can be trained with new signs on the fly with immediate classification. This is something that would be difficult to achieve with traditional HMM approaches. However, the real power of this approach lies in its ability to produce high classification results on ‘one shot’ training and can demonstrate real time training on one individual with successful classification performed on a different individual performing the same signs. Figure 7 shows the word ‘different’ being performed by two different people along with the binary feature vector produced. The similarity is clear, and the signed words are correctly classified.

Acknowledgements. This work is funded by EC Project CogViSys.

References

1. R. Bowden and M. Sarhadi. A non-linear model of shape and motion for tracking finger spelt american sign language. *Image and Vision Computing*, 20(9-10):597–607, 2002.
2. D. B. (ed). *Dictionary of British Sign Language*. British Deaf Association UK, Faber and Faber, ISBN: 0571143466, 1992.
3. S. S. Fels and G. Hinton. Glove-talk: A neural network interface between a data-glove and a speech synthesiser. *IEEE Trans. on Neural Networks*, 4(1):2–8, 1993.
4. M. W. Kadous. Machine recognition of auslan signs using powergloves: towards large lexicon recognition of sign language. In *Proc. Workshop on the Integration of Gesture in Language and Speech*, pages 165–174, 1996.
5. J. Kim, W. Jang, and Z. Bien. A dynamic gesture recognition system for the korean sign language (ksl). *IEEE Trans. Systems, Man and Cybernetics*, 26(2):354–359, 1996.
6. R. Liang and M. Ouhyoung. A real time continuous gesture recognition system for sign language. In *Intl. Conf. on Automatic Face and Gesture Recognition*, pages 558–565, 1998.
7. R. Lockton and A. W. Fitzgibbon. Real-time gesture recognition using deterministic boosting. In *Proc. British Machine Vision Conf.*, 2002.

8. T. Starner and A. Pentland. Visual recognition of american sign language using hidden markov models. In *Intl. Conf. on Automatic Face and Gesture Recognition*, pages 189–194, 1995.
9. B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Filtering using a tree-based estimator. In *Proc. Intl. Conf. on Computer Vision*, volume II, pages 1063–1070, 2003.
10. R. Sutton-Spence and B. Woll. *The Linguistics of British Sign Language, An Introduction*. Cambridge University Press, 1999.
11. C. Vogler and D. Metaxas. Asl recognition based on a coupling between hmms and 3d motion analysis. In *Proc. Intl. Conf. on Computer Vision*, pages 363–369, 1998.
12. C. Vogler and D. Metaxas. Towards scalability in asl recognition: Breaking down signs into phonemes. In *Gesture Workshop*, pages 17–99, 1999.

Appendix: Visual Lexicon Data

Listed here are the 49 signs used in the experiments reported in this paper:

‘I_me’, ‘america’, ‘and’, ‘baby’, ‘because’, ‘british’, ‘but’,
 ‘can’, ‘computer’, ‘deaf’, ‘different’, ‘difficult’, ‘easy’,
 ‘english’, ‘exam_change’, ‘fast’, ‘fingerspelling’, ‘have’,
 ‘hello’, ‘know’, ‘language’, ‘last’, ‘learn’, ‘level’, ‘many’,
 ‘meet’, ‘name’, ‘nice’, ‘people’, ‘recognise’, ‘research’, ‘rich’,
 ‘same’, ‘say’, ‘sign’, ‘start’, ‘summer’, ‘teach’, ‘this’ ‘to’,
 ‘translate’, ‘try’, ‘understand’, ‘want’, ‘we’, ‘what’, ‘wife’,
 ‘with’, ‘yes’.

Fast Object Detection with Occlusions

Yen-Yu Lin¹, Tyng-Luh Liu¹, and Chiou-Shann Fuh²

¹ Inst. of Information Science, Academia Sinica, Nankang, Taipei 115, Taiwan,
liutyng@iis.sinica.edu.tw

² Dept. of CSIE, National Taiwan University, Taipei 106, Taiwan

Abstract. We describe a new framework, based on *boosting algorithms* and *cascade structures*, to efficiently detect objects/faces with occlusions. While our approach is motivated by the work of Viola and Jones, several techniques have been developed for establishing a more general system, including (i) a *robust boosting scheme*, to select useful weak learners and to avoid overfitting; (ii) *reinforcement training*, to reduce false-positive rates via a more effective training procedure for boosted cascades; and (iii) *cascading with evidence*, to extend the system to handle occlusions, without compromising in detection speed. Experimental results on detecting faces under various situations are provided to demonstrate the performances of the proposed method.

1 Introduction

While object detection has long been an important and active area in vision research, most of its applications now demand not only *accuracy* but also (real-time) *efficiency*. Often, to address these two concerns satisfactorily, a typical detection system considers only a certain *regular class* of target objects even though the restriction may limit its practical use. In [17], Viola and Jones propose an effective scheme using *AdaBoost* to detect faces through a *boosted cascade*. Their framework has prompted considerable interest in further investigating the use of boosting algorithms and cascade structures for fast object detection, e.g., [1],[5],[6],[7]. Our detection method also relies on the two elements, but different from the foregoing works, we aim to develop a more general detection system by focusing on the issues of *overfitting* and *occlusion*.

Previous Work. The literature on object detection is quite extensive. We discuss only some of the recent works, especially those based on learning. Also, unless further specified, we focus hereafter on the subject of *face detection*.

Methods based on dimension reduction are often used in detecting faces. Moghaddam and Pentland [9] propose an approach for face detection by calculating several *eigenfaces* from training data. Each detected pattern is then projected into a feature space, formed by the eigenfaces. A testing pattern is classified as face or non-face, depending on its DIFS (Distance-In-Feature-Space) and DFFS (Distance-From-Feature-Space). In [18], Yang et al. develop a face detection system using FLD (Fisher Linear Discriminant), and consider the face classification in a feature space, spanned by the so-called *fisherfaces*.

Sung and Poggio [16] establish a neural network approach that uses a set of face and non-face prototypes to build the hidden layer. The final output is decided by measuring the distances from the detected pattern to each of these prototypes. In [10], Osuna et al. describe an SVM-based method for face detection. They train a hyperplane in some high-dimension feature space for separating faces and non-faces. Each testing pattern is mapped to the feature space for face classification. Romdhani et al. [12] present another SVM-based face detection system by introducing the concept of *reduced set vectors*. Using a sequential evaluation strategy, they report that their face detector is about 15 times faster than the one of Osuna et al. [10]. The SNoW (Sparse Network of Winnows) face detection system by Roth et al. [13] is a sparse network of linear functions that utilizes winnows update rules. They show SNoW is computationally more efficient, and yields better results than those derived by [10].

The excellent work of Viola and Jones [17] has redefined what can be achieved by an efficient implementation of a face detection system. They formulate the detection task as a series of non-face rejection problems. In addition, they calculate an *integral image* to speed up *rectangle feature* computation, and apply AdaBoost to construct stage-wise face classifiers. Since then, a number of systems have been proposed to extend the idea of detecting faces through a boosted cascade. For example, Li et al. [7] develop a system to detect side-view faces by using a coarse-to-fine, simple-to-complex architecture. They divide side-view faces into nine classes, and report that the resulting detector requires about three times computation time than Viola and Jones's to detect the nine kinds of side-view faces. Yet another work by Lienhart and Maydt [5] focuses on extending the set of rectangle features. They rotate extended rectangle features by $\pm 45^\circ$ to obtain rotated rectangle features, and also calculate rotated integral images. In this way, the system can efficiently compute rotated rectangle features by array references. More recently, Liu and Shum [6] introduce a *Kullback-Leibler boosting* to derive weak learners by maximizing projected KL distances. In [1], face patterns in video streams are detected by a boosted cascade, and then classified into different classes of facial expressions. A novel combination of AdaBoost and SVMs (AdaSVMs) is employed so that features selected by AdaBoost are used to form the mapping to a reduced representation for training SVMs.

Our Approach. We generalize the work of Viola and Jones [17] to efficiently detect objects with occlusions. We also deal with the problem of overfitting in training boosted cascades, and thus derive a more robust system. Specifically, the proposed approach leverages its detection performances with three key components. First, we establish a new boosting algorithm that at each iteration, the selection of a weak learner and its coefficient can be determined simultaneously. Each classifier is then formed by a linear combination of the chosen weak learners using a soft-boosting scheme. Second, we propose a reinforcement training procedure to dynamically add difficult and representative training data in each stage. This makes the resulting classifier more general and discriminant. Third, we design a cascading-with-evidence scheme to handle occlusions. The resulting system can detect complete frontal faces and occluded faces at the same time.

2 Classification Using Boosting

Originated from Kearns and Valiant's question [4] to improve the performance of a *weak* learning scheme, boosting has now become one of the most important recent developments in classification methodology. It elegantly leads to a general approach to improve the accuracy of any given learning algorithm. In 1989, Schapire [15] introduces the first polynomial-time boosting procedure. However, it is the AdaBoost [3], proposed by Freund and Schapire, that stimulates the widespread research interest in boosting. A carefully implemented boosting algorithm often gives a compatible performance, with more efficiency, to those yielded by current best classification methods, e.g., SVMs, HMMs.

In this section, we discuss first the ideas of boosting, and then focus on an exponential-loss upper bound on training error. Motivated by [6], we also consider the selection of weak learners by analyzing the weighted projected data. Moreover, we show that an easier-to-implement boosting algorithm can be derived by directly analyzing the error bound, and by addressing overfitting.

2.1 The Ideas behind Boosting

The basic concepts of boosting can be best understood by illustrating with the AdaBoost. Consider now a training set, $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_\ell, y_\ell)\}$, where the first component \mathbf{x} of each sample is the feature value(s), and y is its label. For a two-class classification problem like face detection, $y = 1$ (face) or -1 (non-face), i.e., $D = D^+ \cup D^-$. To elevate the classification performance, AdaBoost uses data re-weighting w_t on D , at iteration t , to iteratively select a weak learner h_t and decide its coefficient α_t in the linear combination of weak learners. It is known that such an iterative process is indeed an attempt to minimize an upper bound of the training error [14]. More precisely, say after T iterations, the training error of a strong classifier $H(\mathbf{x}) = \text{sign}(f(\mathbf{x})) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(\mathbf{x}))$ can be bounded as follows.

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \frac{1}{2} |y_i - H(\mathbf{x}_i)| \leq \frac{1}{\ell} \sum_{i=1}^{\ell} \exp(-y_i f(\mathbf{x}_i)) = \prod_{t=1}^T Z_t, \quad (1)$$

where $Z_t = \sum_{i=1}^{\ell} w_t(i) \exp(-\alpha_t y_i h_t(\mathbf{x}_i))$. At each iteration t , AdaBoost tries to minimize the error bound by reducing Z_t as much as possible via *steepest descent*. When weak learners h_t s are restricted to be binary, it leads to the choice of α_t in [17]. Nevertheless, the relation in (1) still holds for weak learners assuming real values—a *crucial property for selecting good weak learners*.

2.2 Boosting without Overfitting

The foregoing discussion simply points out the two main elements of a boosting algorithm: *weak-learner selection* and *data re-weighting*. For AdaBoost, the new data weight $w_{t+1}(i)$ can be explicitly computed from $w_t(i)$ and α_t . Furthermore,

recent studies suggest that AdaBoost may overfit when the training data contain highly noisy patterns [2], [11]. For face detection via learning, the problem of overfitting is especially delicate and must be handled appropriately in that there are quite a number of non-face patterns resembling faces.

Effective Weak Learner Selection. When efficiency is emphasized, it is preferable to have a classifier of fewer weak learners to achieve the required training accuracy. Meanwhile, the mechanism to select weak learners should take account of its implication on data re-weighting. For instance, the fast detection system of Viola and Jones [17] considers binary weak learners from thresholding on rectangle features. Though their scheme may choose weak learners that are too crude for effectively discriminating the face and non-face distributions, it does have the advantage of using a straightforward updating scheme on data weights w_t , through the analytic form of α_t . On the other hand, the KL boosting of Liu and Shum [6] computes weak learners by maximizing the relative entropy between two 1-D projected distributions of face and non-face samples. At each iteration t , all the coefficients $\alpha_1, \dots, \alpha_t$ for combining the chosen weak learners are re-evaluated and optimized *in parallel*. As a result, the data weights are updated according to heuristic formulas (defined in (8) and (9) of [6]). Motivated by these observations, we describe a method to select useful real-valued weak learners of *positive unit coefficients*, and to conveniently perform data re-weighting iteratively by following the AdaBoost manner.

Assume that we have a set of 1-D mappings $\{\phi_i\}_{i=1}^n$ that each ϕ projects the training data D into real-valued scalars. In our approach, the mapping ϕ will be defined uniquely by a rectangle feature. Thus, we could further assume each ϕ has a compact support. This implies that it is possible to compute histogram distributions for the projected data with a pre-defined partition of m equal-size bins over a finite range of the real line, denoted as $\{b_k\}_{k=1}^m$. Now, focus on how to derive good weak learners with the projected data. Similar to the AdaBoost algorithm used in [17], we try to find, at each iteration t , a weak learner h_t by minimizing Z_t . The differences are: 1) h_t need not be binary, and 2) like [6], each h_t is defined by considering the two-class weighted histograms of projected training data. As our discussion below applies to all iterations, we shall drop the subscript t to simplify the notations.

For each projection ϕ , we define $\mathbf{i}_k(\phi) = \{i \mid \mathbf{x}_i \in D, \phi(\mathbf{x}_i) \in b_k\}$, the indexes of training data being projected by ϕ into bin b_k . Analogously, $\mathbf{i}_k^+(\phi)$ and $\mathbf{i}_k^-(\phi)$ are defined, respectively, for $\mathbf{x}_i \in D^+$ and D^- . With these notations, we are ready to evaluate the values of weighted positive histogram and negative histogram of b_k by

$$p_k^+(\phi) = \sum_{\mathbf{i}_k^+(\phi)} w(i) \quad \text{and} \quad p_k^-(\phi) = \sum_{\mathbf{i}_k^-(\phi)} w(i). \quad (2)$$

Notice that the two weighted histograms p^+ and p^- are not normalized into distributions. Nevertheless, defining them in this way will be more convenient for our analysis, and also without any bearings on the classification outcomes.

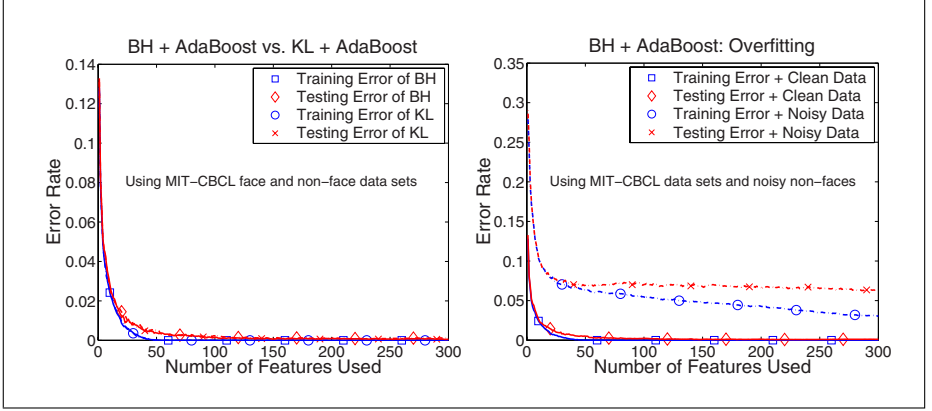


Fig. 1. (a) Using BH or KL weak learners gives a comparable boosting performance. (b) Overfitting is perceived by increasing gaps between training and testing errors.

To establish rules for selecting good weak learners, we consider a projection mapping ϕ and an arbitrary $\mathbf{x} \in D$ such that $\phi(\mathbf{x}) \in b_k$. Let h_ϕ be the weak learner arising from ϕ . Then, it is reasonable to establish the definition of h_ϕ by assuming $h_\phi(\mathbf{x})$ depends on some quantity related to bin b_k . In particular, we write $h_\phi(\mathbf{x}) = s_k$, where s_k is a real scalar. Applying this relation to the definition of Z , we have $Z = \sum_{k=1}^m \sum_{\mathbf{i}_k(\phi)} w(i) \exp(-\alpha y_i s_k)$. Following the strategy of AdaBoost [14], the best choices of s_k , for $k = 1, \dots, m$, can be obtained by minimizing Z with respect to each s_k :

$$\frac{dZ}{ds_k} = 0 \Rightarrow s_k^* = \frac{1}{\alpha} \ln \sqrt{p_k^+(\phi)/p_k^-(\phi)} \quad \text{and} \quad Z^* = 2 \sum_{k=1}^m \sqrt{p_k^+(\phi)p_k^-(\phi)}. \quad (3)$$

The equations in (3) suggest that we can re-scale h_ϕ by multiplying with α such that the selection of weak learner and its coefficient can be decided simultaneously. In addition, the *strength* of a weak learner with respect to the weighted training data can now be measured explicitly with the Bhattacharyya coefficient. We summarize these observations into the following two criteria:

1. Each mapping ϕ implicitly defines a weak learner h_ϕ of coefficient 1 by

$$h_\phi(\mathbf{x}) = \ln \sqrt{p_k^+(\phi)/p_k^-(\phi)}, \quad \text{for all } \mathbf{x} \in D \text{ and } \phi(\mathbf{x}) \in b_k.$$

2. At each iteration, the best weak learner h_ϕ^* is the one that yields the minimal Bhattacharyya coefficient. This result somewhat supports the use of Kullback-Leibler distance in [6]. However, besides a much easier computation, using the Bhattacharyya coefficient has the advantage to skip the estimation of coefficient α , and most of all, it guarantees to optimally minimize Z . In Figure 1a, comparisons between using AdaBoost with the two criteria for selecting weak learners indicate a comparable classification performance.

Algorithm 1: Soft-Boosting with Bhattacharyya weak learners.

Input : A weak learning algorithm *BH-WeakLearn* derived from (3), the number of iterations T , and ℓ labeled training data D .

Output : A strong classifier H .

Initialize the weight vector $w_1(i) = 1/\ell$, for $i = 1, \dots, \ell$.

for $t \leftarrow 1, 2, \dots, T$ **do**

1. Call *BH-WeakLearn*, using distribution w_t on D , to derive $h_t : X \rightarrow \mathbb{R}$.
2. $w_{t+1}(i) \leftarrow w_t(i) \exp(-y_i h_t(\mathbf{x}_i)) / Z_t$, for $i = 1, 2, \dots, \ell$. (Z_t is a normalization factor such that w_{t+1} is a distribution.)

Call $\text{LP}_{\text{REG}}\text{-AdaBoost}$, with inputs $\{y_i h_t(\mathbf{x}_i)\}_{i=1}^T$, to output $\{\beta_t\}_{t=1}^T$.

Output the final strong classifier: $H(\mathbf{x}) = \text{sign}(f(\mathbf{x})) = \text{sign}(\sum_{i=1}^T \beta_i h_t(\mathbf{x}))$.

Table 1. Error Rate: Soft-Boosting against different degree of noisy non-face data.

# of h_t	No Noise	25% Noise		50% Noise		75% Noise		100% Noise	
	BH	BH	BH+Soft	BH	BH+Soft	BH	BH+Soft	BH	BH+Soft
100	0.23	2.96	2.36	4.91	4.07	5.50	4.54	7.03	5.85
200	0.12	2.29	1.99	4.11	3.41	5.16	3.91	6.65	5.23
300	0.11	2.13	1.85	3.77	2.92	5.23	3.74	6.39	4.97

Soft-Boosting. Rätsch et al. [11], show that AdaBoost behaves asymptotically like a hard-margin classifier, and propose to use soft margins for AdaBoost (or *soft-boosting* for abbreviation) to avoid data overfitting (see Figure 1b). Extending AdaBoost to incorporate soft margins allows certain of difficult patterns to be misclassified within some ranges during the training stage. Such a strategy has been tested extensively and successfully in SVMs. We adopt the $\text{LP}_{\text{REG}}\text{-AdaBoost}$ in [11] that uses linear programming with slack variables to achieve soft margins. The $\text{LP}_{\text{REG}}\text{-AdaBoost}$ is paired nicely with our implementation in that it only needs the T weak learners and the margin distributions as inputs. The coefficients of weak learners are not used at all. Since the resulting weak learners have positive unit coefficients, all the available information from the training stage for selecting weak learners is passed to the regularized linear programming. In Table 1, we summarize experimental results of applying soft-boosting to deal with overfitting, using the MIT-CBCL face and non-face data sets. The noisy non-face data are generated from the false positive samples at different stages of a cascade structure, proposed by Viola and Jones [17]. When combining the Bhattacharyya weak learners with soft-boosting, consistent improvements in the detection rates are obtained in all experiments. A complete description of our method is listed in Algorithm 1.

3 Fast Detection with Occlusions

We construct a feature-based face detection system through a *simple-to-complex* cascade structure. Such a strategy reduces a face detection problem into the rejections of non-face patterns stage-wise. As it turns out, to detect occluded

faces via a boosted cascade is a nontrivial problem because they are very likely to be rejected at some intermediate stage. While adopting a more lenient policy in rejecting non-face patterns may be able to handle occlusions to some degree, it often causes an increase in the false-positive detection rate. In dealing with these issues, we propose a method that consists of two schemes: 1) *reinforcement training*, to reduce the false positive rates, and 2) *cascading with evidence*, to detect faces with occlusions.

3.1 Reinforcement Training

In a cascade structure \mathcal{M} , the face detector at stage k can be denoted as $H_k(\mathbf{x}) = \text{sign}(f_k(\mathbf{x})) = \text{sign}(\sum_{t=1}^{T_k} \beta_t h_t(\mathbf{x}))$, where the size of T_k increases as k becomes large. We also use \mathcal{M}_1^k to represent the *sub-cascade* of the first k stages. To train a cascade of face detectors, we usually start with a balanced two-class data D , i.e., the same number of face and non-face data. During training, data arriving at the k th stage are those that pass the sub-cascade \mathcal{M}_1^{k-1} . Among them, most are face data and only a few are non-face. The situation can lead to a problem that there could be too few non-face data to be trained with, and consequently, a boosting method may yield unreliable decision boundary predictions. To alleviate this problem, we collect an additional set \mathcal{N} of images that do not contain any face patterns, and then perform a two-step reinforcement training whenever there are too few non-face samples reaching some stage k .

1. The *Bootstrap* technique is applied to generate non-face patterns by testing all images in \mathcal{N} with the sub-cascade \mathcal{M}_1^{k-1} . Those that survive are indeed false positives of \mathcal{M}_1^{k-1} . We denote them as \mathcal{N}_k .
2. In practice, when k is small, there are way too many samples of \mathcal{N}_k to be considered. For each $\mathbf{x} \in \mathcal{N}_k$, the mapping $\mathbf{x} \mapsto (\tilde{f}_1(\mathbf{x}), \dots, \tilde{f}_{k-1}(\mathbf{x}))$ is used to associate \mathbf{x} with a $(k-1)$ -dimension feature vector. (Note that each \tilde{f} is normalized from the stage-wise f such that $\sum_{\mathbf{x} \in \mathcal{N}_k} \tilde{f}(\mathbf{x}) = 1$.) Then, *k-means clustering* is applied to divide \mathcal{N}_k into six clusters [16] to model the empirical non-face distribution. The needed non-face samples, at each stage k , can now be selected uniformly from the six clusters.

The reinforcement strategy enables the system to consider meaningful and difficult non-face samples from \mathcal{N} . Though it is difficult to model the distribution of non-face patterns, we establish an effective way in selecting representative ones. With reinforcement training, the system is expected to have a lower false positive rate in each cascade stage of a testing procedure.

3.2 Cascading with Evidence

Mentioned briefly in Section 2.2, rectangle features used in [17] can be computed rather efficiently by referencing integral images. A rectangle feature simply computes the difference between sums of pixel intensity in adjacent regions, and hence a strong classifier formed by these rectangle features records intensity distribution in the faces. Apparently, detection systems using rectangle features are

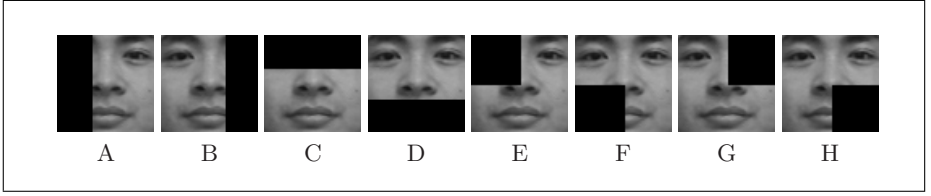


Fig. 2. From left to right: eight types of occluded faces (1/3 or 1/4 occlusions).

sensitive to occlusions because the intensity differences related to occluded regions are no longer reliable. We instead treat each rectangle feature as a mapping ϕ that projects $\mathbf{x} \in D$ to the resulting intensity difference, and then compute the Bhattacharyya coefficient between the weighted positive and negative histograms. Still, the derived classifiers only work for *regular* faces—to account for occlusions, other mechanisms are needed.

To distinguish an occluded face from non-face samples, the clues lie in the *evidence* left behind when a testing sample is in question. In Figure 2, there are eight types of occlusions that our detection system is designed to handel. The face data we choose to train our system are images of size 20×20 . Totally, about 80,000 rectangle features can be generated, and represented as a set Ψ . (We use the same three types of rectangle features proposed in [17].) Then, for each type \mathcal{I} of occluded faces shown in Figure 2, we use $\mathcal{O}_{\mathcal{I}}$ to denote the occluded region, and define the largest subset of Ψ disjoint from $\mathcal{O}_{\mathcal{I}}$ by

$$\Psi_{\mathcal{I}} = \{\psi \mid \psi \in \Psi \text{ and } \psi \cap \mathcal{O}_{\mathcal{I}} = \emptyset\}, \quad \text{for } \mathcal{I} = \mathcal{A}, \mathcal{B}, \dots, \mathcal{H}. \quad (4)$$

Now, in testing a sample \mathbf{x} at some stage k of the cascade, besides calculating $H_k(\mathbf{x})$, we also compute an additional eight-dimensional feature vector, the evidence of \mathbf{x} at stage k , defined as follows:

$$\mathcal{E}_k(\mathbf{x}) = (f_k^{\mathcal{A}}(\mathbf{x}), f_k^{\mathcal{B}}(\mathbf{x}), \dots, f_k^{\mathcal{H}}(\mathbf{x})) \quad \text{and} \quad f_k^{\mathcal{I}}(\mathbf{x}) = \sum_{\mathcal{T}} \beta_{\mathcal{T}} h_{\mathcal{T}}(\mathbf{x}), \quad (5)$$

where the summation $\sum_{\mathcal{T}}$ involves only those weak learners that their corresponding rectangle features do not intersect with $\mathcal{O}_{\mathcal{I}}$, for $\mathcal{I} = \mathcal{A}, \mathcal{B}, \dots, \mathcal{H}$. With (5), we propose a *cascading with evidence* scheme to detect faces with the eight types of occlusions efficiently, where its advantages are summarized below.

- Since each $\beta_{\mathcal{T}} h_{\mathcal{T}}(\mathbf{x})$ has already been evaluated in the computation of $H_k(\mathbf{x})$, the evidence vector $\mathcal{E}_k(\mathbf{x})$ is easier to derive.
- To illustrate, let \mathbf{x} be a face sample of type- \mathcal{A} occlusion, and \mathbf{x} is being considered to be rejected as a non-face pattern due to $H_k(\mathbf{x}) < 0$. Then, we can reference its evidence vectors from the k stages. In particular, the majority of $f_1^{\mathcal{A}}, \dots, f_k^{\mathcal{A}}$ should be positive responses to indicate \mathbf{x} is a type- \mathcal{A} occluded face. Such a property is not shared by most true non-face samples. The details of cascading with evidence are given in Algorithm 2 and 3.

Algorithm 2: Cascading with Evidence: Training Procedure

-
- Input** : Rectangle feature sets, Ψ and $\Psi_{\mathcal{I}}$, for $\mathcal{I} = \mathcal{A}, \mathcal{B}, \dots, \mathcal{H}$.
Output : A main cascade \mathcal{M} , and 8 occlusion cascades, $\mathcal{I} = \mathcal{A}, \mathcal{B}, \dots, \mathcal{H}$.
1. Train a regular cascade \mathcal{V} from Ψ , using the techniques in [17].
 2. Train 8 occlusion cascades \mathcal{I} from $\Psi_{\mathcal{I}}$, using \mathcal{V} as a benchmark.
 3. $T_k^{\mathcal{I}} \leftarrow$ Number of weak learners used at the k th stage of \mathcal{I} .
 4. Train cascade \mathcal{M} such that $|\{h_1, \dots, h_{T_k}\} \cap \Psi_{\mathcal{I}}| \geq T_k^{\mathcal{I}}$, for each stage k .
-

Algorithm 3: Cascading with Evidence: Testing Procedure

-
- Input** : A testing pattern, \mathbf{x} .
Output : Face, Non-Face, or Type- \mathcal{I} Occluded Face.
1. If \mathbf{x} goes through \mathcal{M} return **Face**.
 2. If \mathbf{x} is rejected at stage k and all $f_k^{\mathcal{I}} < 0$ return **Non-Face**.
 3. Dispatch \mathbf{x} to cascade \mathcal{I} if $f_k^{\mathcal{I}} > 0$ and $\sum_{t=1}^k f_t^{\mathcal{I}}$ is the largest.
- if** \mathbf{x} goes through \mathcal{I} **then**
 └ return Type- \mathcal{I} Occluded Face
else
 └ return Non-Face
-

4 Experimental Results

The face training data are obtained from MIT-CBCL database and AR [8] face database. They are pictured under different lighting, facial expressions, and poses. We rotate ($\pm 15^\circ$) and mirror each face image, and crop the face region with slightly different scales. The non-face training data are collected from the Internet. Both face and non-face training data are resized at the resolution, 20 by 20 pixels. Totally, 10,000 face images and 10,000 non-face images are used as our initial training data. We also prepare about 16,000 images that contain no faces for generating non-face training data in reinforcement training.

A rectangle feature is indeed a Haar wavelet filter that maps each training image into a real value. Thus, each rectangle feature gives rise to two different distributions for face and non-face data. When the number of training samples is fixed, the number of bins used to model the two distributions is decided on the tradeoff between accuracy and data overfitting. Empirically, we have used 10 bins to derive satisfactory results. At each stage, we aim to construct a face detector with a loose threshold by reducing the false positive rate, while detecting almost all positive samples. This is achieved by adding weak learners until the false positive rate is less than 40%, and also the detection rate is higher than 99.9%.

In our implementation, a regular cascade to detect frontal faces (without occlusion) has 21 stages, including 872 weak learners. The first three stages contain 2, 3, and 5 weak learners respectively. We test it on the benchmark testing set, i.e., CMU+MIT data set. It contains 130 images with 507 frontal faces. There are totally 81,519,506 sub-windows scanned. The detecting results are represented as an ROC curve shown in Figure 3b. The performance is comparable to other

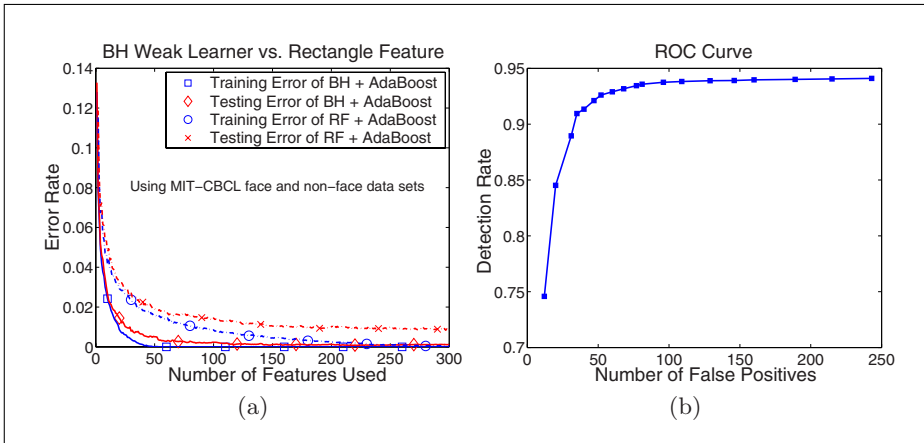


Fig. 3. (a) Using Bhattacharyya weak learners outperforms implementation with rectangle features in accuracy and convergence speed. (b) The ROC curve of our detector performed on CMU+MIT data set (81, 519, 506 sub-windows scanned).

Table 2. Stage Passing Rate of a Cascade Using CMU+MIT Data Set

Stage	Passing Rate	Stage	Passing Rate	Stage	Passing Rate	Stage	Passing Rate
1	0.35503	4	0.01254	7	0.00082	10	0.00018
2	0.10934	5	0.00383	8	0.00048	11	0.00013
3	0.04611	6	0.00165	9	0.00023	12	0.00009

existing face detectors, e.g., [16], [17]. The stage-wise classification efficiency is shown in Table 2. The first stage rejects about 64.5% sub-windows (almost non-face). Averagely, a sub-window is classified by only 4.59 weak learners. In addition, the advantages of using Bhattacharyya weak learners in improving accuracy and convergence speed are also illustrated in Figure 3a.

To detect frontal and occluded faces at the same time, we need a main cascade, \mathcal{M} , and eight occlusion cascades. In designing \mathcal{M} , besides satisfying the detection rate constraints mentioned above, at each stage, the number of weak learners used should satisfy the condition described in Algorithm 2-(4). The requirements are to ensure that every component of the evidence vector \mathcal{E}_k is well-defined at each stage k . Since a weak learner may be associated with more than one type. The number of weak learners used in each stage of the main cascade \mathcal{M} is still manageable to produce efficient detection. (The first three stages of \mathcal{M} contain 7, 9, and 12 weak learners, respectively.) Regarding the eight occlusion cascades, each of them contains from 23 to 26 stages respectively. Applying cascading with evidence, we detect frontal and eight kinds of occluded faces in *three times* computing time used in detecting only frontal faces. It detects about 18 320x240 frames per second on a P4 3.06GHz PC. A number of experimental results are reported in Figure 4 to demonstrate the effectiveness of our system.



Fig. 4. (a)-(f) Detection results derived by applying our face detector to some of the CMU+MIT data set. (g)-(l) Detection results on several occluded faces. (m)-(o) Detection results for various exaggerated expressions.

References

1. Bartlett, M.S., Littlewort, G., Fasel, I., Movellan, J.R.: Real time face detection and facial expression recognition: Development and applications to human computer interaction. In: Computer Vision and Pattern Recognition HCI Workshop, Madison, Wisconsin, USA (2003)
2. Dietterich, T.G.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning* **40** (2000) 139–157
3. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Proc. European Conf. Computational Learning Theory, Barcelona, Spain (1995) 23–37
4. Kearns, M., Valiant, L.G.: Learning boolean formulae or finite automata is as hard as factoring. Technical report, Harvard University Aiken Computation Laboratory (1988)
5. Lienhart, R., Maydt, J.: An extended set of Haar-like features for rapid object detection. In: Proc. Int'l Conf. Image Processing. Volume 1., Rochester, NY, USA (2002) 900–903
6. Liu, C., Shum, H.: Kullback-Leibler boosting. In: Proc. Conf. Computer Vision and Pattern Recognition. Volume 1., Madison, Wisconsin, USA (2003) 587–594
7. Li, S., Zhu, L., Zhang, Z., Blake, A., Zhang, H., Shum, H.: Statistical learning of multi-view face detection. In: Proc. Seventh European Conf. Computer Vision. Volume 4., Copenhagen, Denmark (2002) 67–81
8. Martinez, A., Benavente, R.: The AR face database. Technical report, CVC Technical Report #24 (1998)
9. Moghaddam, B., Pentland, A.P.: Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19** (1997) 696–710
10. Osuna, E., Freund, R., Girosi, F.: Training support vector machines: An application to face detection. In: Proc. Conf. Computer Vision and Pattern Recognition, San Juan, Puerto Rico (1997) 130–136
11. Rätsch, G., Onoda, T., Müller, K.R.: Soft margins for AdaBoost. *Machine Learning* **42** (2001) 287–320
12. Romdhani, S., Torr, P., Schölkopf, B., Blake, A.: Computationally efficient face detection. In: Proc. Eighth IEEE Int'l Conf. Computer Vision. Volume 2., Vancouver, BC, Canada (2001) 695–700
13. Roth, D., Yang, M., Ahuja, N.: A SNoW-based face detector. In: Advances in Neural Information Processing Systems, Denver, CO, USA (2000) 855–861
14. Schapire, R.E., Singer, Y.: Improved boosting using confidence-rated predictions. *Machine Learning* **37** (1999) 297–336
15. Schapire, R.E.: The strength of weak learnability. *Machine Learning* **5** (1990) 197–227
16. Sung, K.K., Poggio, T.: Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20** (1998) 39–51
17. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. Conf. Computer Vision and Pattern Recognition. Volume 1., Kauai, HI, USA (2001) 511–518
18. Yang, M.H., Ahuja, N., Kriegman, D.: Face detection using mixtures of linear subspaces. In: Proc. Int'l Conf. Automatic Face and Gesture Recognition, Grenoble, France (2000) 70–76

Pose Estimation of Free-Form Objects

Bodo Rosenhahn¹ and Gerald Sommer²

¹ University of Auckland (CITR)
Computer Science Department
Tamaki Campus
Private Bag 92019 Auckland
New Zealand
`bros028@cs.auckland.ac.nz`

² Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40, 24098 Kiel, Germany
`gs@ks.informatik.uni-kiel.de`

Abstract. In this contribution we present an approach for 2D-3D pose estimation of 3D free-form surface models. In our scenario we observe a free-form object in an image of a calibrated camera. Pose estimation means to estimate the relative position and orientation of the 3D object to the reference camera system. The object itself is modeled as a two-parametric 3D surface and extended by one-parametric contour parts of the object. A twist representation, which is equivalent to a Fourier representation allows for a low-pass approximation of the object model, which is advantageously applied to regularize the pose problem. The experiments show, that our developed algorithms are fast (200ms/frame) and accurate (1° rotational error/frame).

1 Introduction

Pose estimation itself is one of the oldest computer vision problems. It is crucial for many computer and robot vision tasks. Pioneering work was done in the 80's and 90's by Lowe [8], Grimson [7] and others. These authors use point correspondences. More abstract entities can be found in [16,3]. In the literature we find circles, cylinders, kinematic chains or other multi-part curved objects as entities. Works concerning free-form curves can be found e.g. in [5]. Contour point sets, affine snakes, or active contours are used for visual servoing in different works. For a definition of the pose problem we want to quote Grimson [7]: *By pose we mean the transformation needed to map an object model from its inherent coordinate system into agreement with the sensory data.* We are estimating the relative rotation and translation of a 3D object with respect to a reference camera system in the framework of a 2D-3D pose estimation approach. In this work we deal with free-form surface and contour models for object representation. We want to quote Besl [2] for a definition: *A free-form surface has a well defined surface that is continuous almost everywhere except at vertices, edges and cusps.*

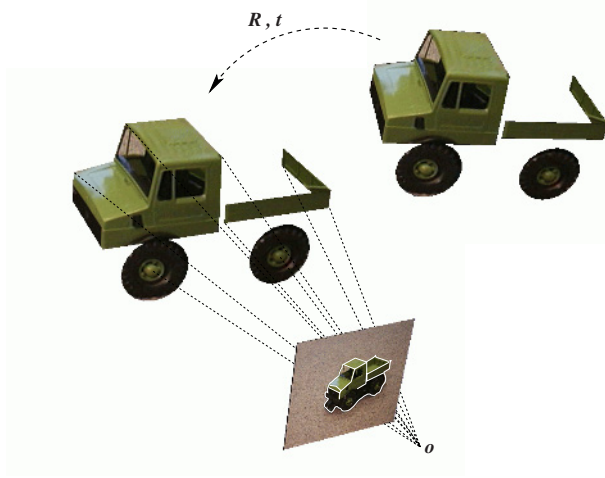


Fig. 1. The scenario. The assumptions are the projective camera model, the model of the object, extracted features and the silhouette of the object on the image plane. The aim is to find the pose (\mathbf{R}, \mathbf{t}) of the model, which leads to the best fit of the object with the image data.

In section 2 we start with a summary of our preliminary works regarding feature based and contour based pose estimation [12]. Then we present our extensions in section 3: An approach for silhouette based pose estimation of free-form surface models. In this approach we assume an extracted image contour of the observed object model. Only by using this contour we estimate the pose of the surface model with respect to a calibrated reference camera system. The surface model is parametrically represented based on a signal model from which low-pass approximations are derived.

It is clear, that only using the image contour results in a loss of further available object information in the image plane. Therefore, we will also present an extension which takes into account additionally available object information. We are using additional contour parts of the object which are brought to correspondence with extracted image features inside the object silhouette.

In the experiments, section 4, we present pose results of objects which are tracked successfully, even with noisy extracted image contours. We will show that our algorithms are able to cope with occlusions caused by the motion and to compensate errors to some degree. The contribution ends with a discussion in section 5.

To deal with geometric aspects of the pose problem, we use as mathematical language so-called Clifford or geometric algebras [14]. Here we will give no theoretical introduction into the concepts of Clifford algebras but want to point out a few properties which are important for this problem: The elements in geometric algebras are called multivectors which can be multiplied by using a geometric product. In geometric algebra Euclidean, projective and conformal geometry [9] find the frame where they can reconcile and express their respective potential. Besides, it enables a coordinate-free and dense symbolic representation. To model

the pose problem, we use the conformal geometric algebra (CGA). The CGA is build up on a conformal model (geometry on the sphere) which is coupled with a homogeneous model to deal with kinematics and projective geometry simultaneously. This enables us to deal with the Euclidean, kinematic and projective space in one framework and therefore to cope with the pose problem in an efficient manner. Furthermore the unknown rigid motions are expressed as so-called *motors* which can be applied on different entities (e.g. points or lines) by the use of the geometric product. This leads to compact and easily interpretable equations. In the equations we will use the inner product, \cdot , the outer product, \wedge , the commutator, \times , and anticommutator, $\overline{\times}$, product, which can be derived from the geometric product. Though we will also present equations formulated in conformal geometric algebra, we only explain these symbolically and want to refer to [12] for more detailed information.

2 Preliminary Work

We start with a few aspects of our preliminary works which build the basis for this contribution. First, we will present point based pose estimation and then we continue with the approach for contour based free-form pose estimation.

2.1 Point Based Pose Estimation

For 2D-3D point based pose estimation we are using constraint equations which compare 2D image points with 3D object points. The use of points is the simplest representation for 3D objects treated here. To compare a 2D image point \mathbf{x} with 3D object points $\underline{\mathbf{X}}$, the idea is to reconstruct from the image point a 3D projection ray, $\underline{\mathbf{L}}_x = \mathbf{e} \wedge (\mathbf{O} \wedge \mathbf{x})$, as Plücker line [10]. The motor \mathbf{M} as exponential of a twist, Ψ , $\mathbf{M} = \exp(-\frac{\theta}{2}\Psi)$, formalizes the unknown rigid motion as a screw motion [10]. The motor \mathbf{M} is applied on the object point $\underline{\mathbf{X}}$ as versor product, $\underline{\mathbf{X}}' = \mathbf{M}\underline{\mathbf{X}}\widetilde{\mathbf{M}}$, where $\widetilde{\mathbf{M}}$ represents the so-called reverse of \mathbf{M} . Then the rigidly transformed object point, $\underline{\mathbf{X}}'$, is compared with the reconstructed line, $\underline{\mathbf{L}}_x$, by minimizing the error vector between the point and the line. The representation of such a constraint equation takes in geometric algebra the form

$$\underbrace{\underbrace{(\mathbf{M} \quad \underline{\mathbf{X}} \quad \widetilde{\mathbf{M}})}_{\text{object point}} \quad \times \quad \underbrace{\mathbf{e} \wedge (\mathbf{O} \wedge \mathbf{x})}_{\text{projection ray, reconstructed from the image point}}}_{\text{rigid motion of the object point} \quad \text{collinearity of the transformed object point with the reconstructed line}} = 0.$$

Note, that we work with a 3D formalization of the pose problem. The constraint equations can be solved by linearization (this means solving the equations for the twist-parameters which generate the screw motion) and by applying the Rodrigues formula for reconstruction of the group action [10]. Iteration leads to a gradient descent method in 3D space. This is more detailed presented in [12].

There we also introduce similar equations to compare 3D points with 2D lines (3D planes) and 3D lines with 2D lines (3D planes). The pose estimation can be performed in real-time and we need 2ms to estimate a pose containing 100 point correspondences on a Linux 2GHz machine.

2.2 Contour Based Pose Estimation

Though point concepts or higher order features are often used for pose estimation [3,16], there exist certain scenarios (e.g. in natural environments), where it is not possible to extract features like corners or curve segments, but just general contours. Therefore we are interested in modeling free-form objects and embedding them into the pose problem.

Fourier descriptors can be used for object recognition [6] and affine invariant pose estimation [1] of closed contours. They have the advantage of a low-pass object representation (as explained later) and they interpolate sample points along a contour as a continuously differentiable function. During our research we rediscovered the use of Fourier descriptors since they are the generalization of so-called *twist-generated* curves we used to model cycloidal curves (cardioids, nephroids etc.) within the pose problem [12]. We now deal with the representation of 3D free-form contours in order to combine these with our previously introduced point based pose estimation constraints. Since the later introduced pose estimation algorithm for surface models goes back to a contour based one, the recapitulation of our former works on contour based pose estimation is of importance.

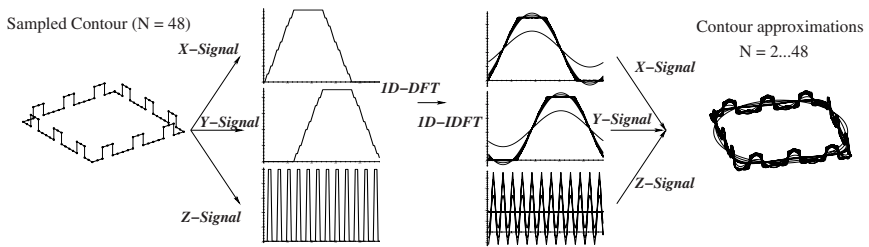


Fig. 2. Visualization of contour modeling and approximation by using three 1D Fourier transformations.

The main idea is to interpret a 1-parametric 3D closed curve as three separate 1D signals which represent the projections of the curve along the x , y and z axis, respectively. Since the curve is assumed to be closed, the signals are periodic and can be analyzed by applying a 1D discrete Fourier transform (1D-DFT). The inverse discrete Fourier transform (1D-IDFT) enables to reconstruct low-pass approximations of each signal. Subject to the sampling theorem, this leads to the representation of the 1-parametric 3D curve $C(\phi)$ as

$$C(\phi) = \sum_{m=1}^3 \sum_{k=-N}^N p_k^m \exp\left(\frac{2\pi k\phi}{2N+1} l_m\right).$$

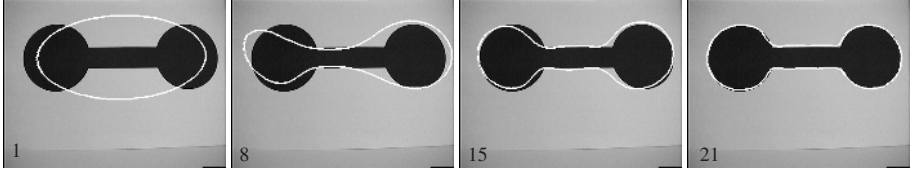


Fig. 3. Pose results of the low-pass filtered contour during the iteration.

The parameter m represents each dimension and the vectors \mathbf{p}_k^m are phase vectors obtained from the 1D-DFT acting on dimension m . In this equation we have replaced the imaginary unit $i = \sqrt{-1}$ with three different rotation planes, represented by the bivectors \mathbf{l}_i , with $\mathbf{l}_i^2 = -1$. Using only a low-index subset of the Fourier coefficients results in a low-pass approximation of the object model which can be used to regularize the pose estimation algorithm. The principle of modeling free-form contours is visualized in figure 2.

For pose estimation this model is then combined with a version of an ICP-algorithm [15]. Figure 3 shows an example. As can be seen, we refine the pose results by adding successively higher frequencies to a low-pass approximation during the iteration. This is basically a multi-resolution method and helps to avoid local minima during the iteration.

3 Surface Based Pose Estimation

After this recapitulation we will now present the main ideas for surface based pose estimation. We start with the extension of the 3D contour model to a 3D surface model and present the basic pose estimation algorithm for free-form surfaces [13]. Then we will continue with extensions of this approach.

3.1 Surface Representation

We are now concerned with the formalization of surfaces in the framework of 2D Fourier descriptors. This will enable us to regularize the estimation and to refine the object model during iteration steps. Hence the multi-scale object representation can be adapted to its inherent geometric complexity. We assume a two-parametric surface [4] of the form

$$F(\phi_1, \phi_2) = \sum_{i=1}^3 f^i(\phi_1, \phi_2) \mathbf{e}_i.$$

This means, we have three 2D functions $f^i(\phi_1, \phi_2) : \mathbb{R}^2 \rightarrow \mathbb{R}$ acting on the different Euclidean base vectors \mathbf{e}_i ($i = 1, \dots, 3$). The idea behind a two-parametric surface is to assume two independent parameters ϕ_1 and ϕ_2 to sample a 2D surface in 3D space. Projecting this function along \mathbf{e}_1 , \mathbf{e}_2 and \mathbf{e}_3 leads to the three 2D functions $f^i(\phi_1, \phi_2)$. For a discrete number of sampled points, f_{n_1, n_2}^i , ($n_1 \in [-N_1, N_1]$; $n_2 \in [-N_2, N_2]$; $N_1, N_2 \in \mathbb{N}$, $i = 1, \dots, 3$) on the surface, we

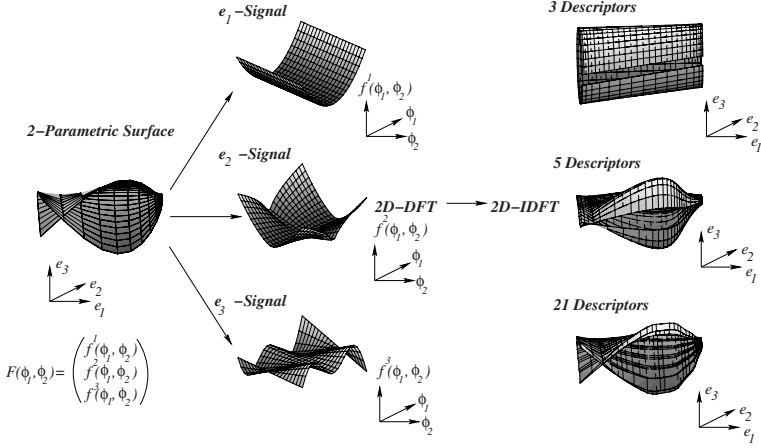


Fig. 4. Visualization of surface modeling and approximation by using three 2D Fourier transformations.

can now interpolate the surface by using a 2D discrete Fourier transform (2D-DFT) and then apply an inverse 2D discrete Fourier transform (2D-IDFT) for each base vector separately. Subject to the sampling theorem the surface can be written as a Fourier representation which appears in geometric algebra as

$$\begin{aligned}
 F(\phi_1, \phi_2) &= \sum_{i=1}^3 \sum_{k_1=-N_1}^{N_1} \sum_{k_2=-N_2}^{N_2} p_{k_1, k_2}^i \exp\left(\frac{2\pi k_1 \phi_1}{2N_1 + 1} l_i\right) \exp\left(\frac{2\pi k_2 \phi_2}{2N_2 + 1} l_i\right) \\
 &= \sum_{i=1}^3 \sum_{k_1=-N_1}^{N_1} \sum_{k_2=-N_2}^{N_2} R_{1,i}^{k_1, \phi_1} R_{2,i}^{k_2, \phi_2} p_{k_1, k_2}^i \widetilde{R_{2,i}^{k_2, \phi_2}} \widetilde{R_{1,i}^{k_1, \phi_1}}.
 \end{aligned}$$

The complex Fourier coefficients are contained in the vectors p_{k_1, k_2}^i that lie in the plane spanned by l_i . We will again call them phase vectors. These vectors can be obtained by a 2D-DFT of the sample points f_{n_1, n_2}^i on the surface,

$$\begin{aligned}
 p_{k_1, k_2}^i &= \frac{1}{(2N_1 + 1)(2N_2 + 1)} \\
 &\quad \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} f_{n_1, n_2}^i \exp\left(-\frac{2\pi k_1 n_1}{2N_1 + 1} l_i\right) \exp\left(-\frac{2\pi k_2 n_2}{2N_2 + 1} l_i\right) e_i.
 \end{aligned}$$

This is visualized in figure 4 as extension to the 1D case of figure 2: a two-parametric surface can be interpreted as three separate 2D signals interpolated and approximated by using three 2D-DFTs and 2D-IDFTs, respectively.

3.2 Silhouette Based Pose Estimation of Free-Form Surfaces

We now continue with the algorithm for silhouette based pose estimation of surface models. In our scenario, we assume to have extracted the silhouette

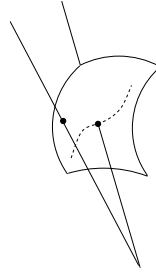


Fig. 5. A main problem during pose estimation of surface models: There is need to express tangentiality between the surface and the reconstructed projection rays. Pure intersection is not sufficient for pose estimation.

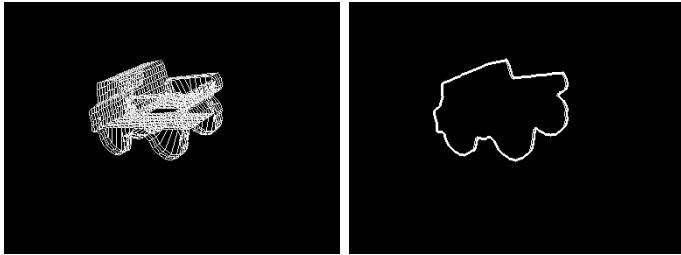


Fig. 6. Left: The surface model projected on a virtual image. Right: The estimated 3D silhouette of the surface model, back projected in an image.

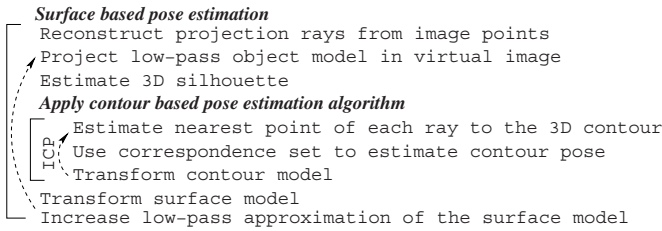


Fig. 7. The algorithm for pose estimation of surface models.

of an object in an image. In the experiments this is simply done by using a thresholded color interval and by smoothing the resulting binary image with morphological operators, see figure 10.

A main problem is, that it is not useful to express an intersection constraint between the reconstructed projection rays and the surface model. This is visualized in figure 5: Postulating the intersection of rays with the surface leads to the effect, that the object is moved directly in front of the camera. Then every reconstructed ray intersects the surface and the constraint is trivially fulfilled. Therefore there is need to express tangentiality between the surface and the reconstructed projection rays and there is need to express a distance measure within our description.

To solve this problem we propose to get from the surface model to a contour model which is tangential with respect to the camera coordinate system. To compare points on the image silhouette with the surface model, the idea is to work with those points on the surface model which lie on the outline of a 2D projection of the object. This means we work with the 3D silhouette of the surface model with respect to the camera. To obtain this, we project the 3D surface on a virtual image. Then the contour is calculated and from the image contour the 3D silhouette of the surface model is reconstructed. This is visualized in figure 6. The contour model is then applied on our previously introduced contour based pose estimation algorithm. Since the aspects of the surface model are changing during the ICP-cycles, a new silhouette will be estimated after each cycle to deal with occlusions within the surface model. The algorithm for pose estimation of surface models is summarized in figure 7.

Note, this approach can easily be extended to a multiple-component silhouette based pose estimation algorithm: If an object consists of several rigidly coupled surface patches, still one 3D contour can be estimated from the including free-form parts and applied to the pose estimation algorithm. This is presented in section 4.2.

3.3 Combining Contour and Surface Patches

We will now present a mixed-mode approach which applies additional edge information on the silhouette based pose estimation. We call additional edges, which are not on the outline of the surface contour with respect to the camera, 'internal' edges, since they are inside the boundary contour in the image. Depending on the object, they can be easily obtainable features which we want to use as additional information to stabilize the pose result. This means to extend the assumed model from one 3D component to multiple components of different dimension. These additional components are representing parts of contours within the outer silhouette of the object. To obtain 'internal' edge information we perform the following image processing steps:

1. Back-ground subtraction from the object
2. Laplace filtering and subtraction of the contour from the filtered image
3. Sub-sampling

This is visualized in figure 8: The first image shows the back-ground subtraction from the object. After this we filter the image and estimate an internal edge image as shown in the second image. The third image shows the sub-sampling to obtain a number of *internal* points we use for pose estimation.

It is useless to claim incidence of these extracted points with one given surface model since they do not contribute any information on the pose quality (as discussed in section 3.2). Instead, within the mixed-mode model of multiple components, their contribution to the pose estimation results increases the accuracy and the robustness with respect to occlusions. The generated set of equations

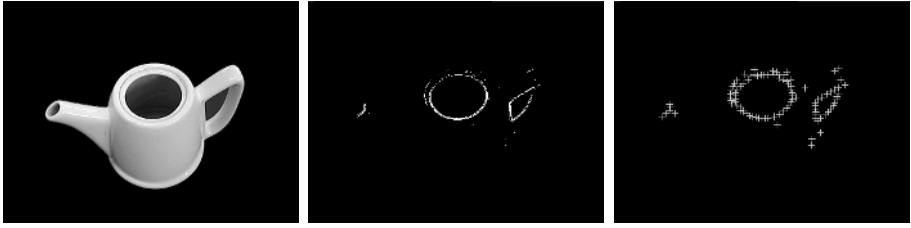


Fig. 8. Image processing steps for getting internal object features: Background subtraction and internal edge detection.

can now be separated in two parts, those obtained from the silhouette and those obtained from the internal feature points. Since both parts can contain larger mismatches or wrong correspondences (see e.g. the falsely extracted edges in figure 8), an outlier elimination is applied to reduce wrong correspondences [12].

4 Experiments

In the experiments we will start with results obtained from a pure silhouette based pose estimation of single patch surface models and continue with the use of multiple surface patches. In section 4.3 we will then deal with pose estimation using additional internal object features.

4.1 One-Component Silhouette Based Pose Estimation

The convergence behavior of the silhouette based pose estimation algorithm is shown in figure 9. As can be seen, we refine the pose results by adding successively higher frequencies during the iteration. This is basically a multi-resolution method and helps to avoid getting stuck in local minima during the iteration. The aim of the first experiment is not only to visualize the pose results, but also to compare the pose results with a ground truth: We put a car model on a turn table and perform a 360 degrees rotation. We further assume the Euclidean 3D surface model of the car and a calibrated camera system observing the turn table. The rotation on the turn table corresponds to a 360 degrees rotation around the y -axis in the calibrated camera system. During the image sequence we apply the silhouette based free-form pose estimation algorithm. Example images (and pose results) of this sequence with extracted image silhouettes are shown in figure 10. As can be seen, there exist shadows under the car, which lead to noisy segmented images. Also some parts of the car (e.g. the front bumper or the tow coupling) are not exactly modeled. This results in errors which are detected during pose estimation. After the detection of failure correspondences they are eliminated in the generated system of equations and they do not influence the result of the pose. Figure 11 shows the absolute error of the estimations in degrees during the whole image sequence. In the image sequence, the maximum error is 5.73 degrees (at the beginning of the sequence). The average absolute

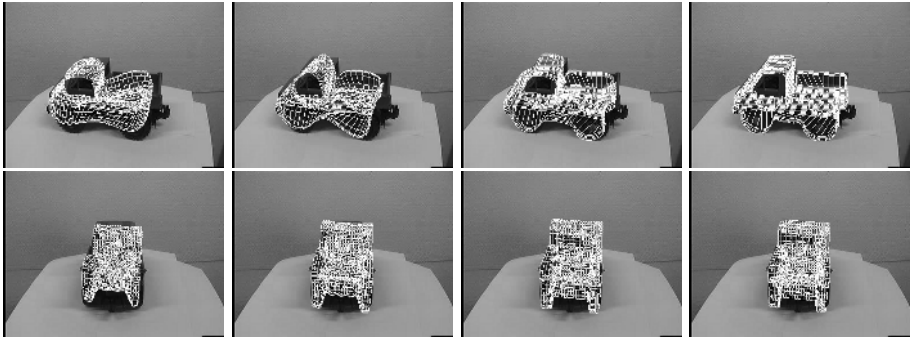


Fig. 9. Pose results of the low-pass contours during the ICP cycle.

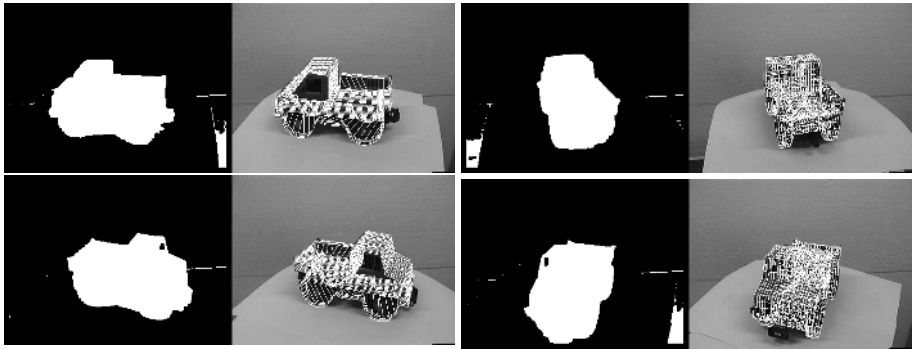


Fig. 10. Pose results of the car model on a turn table and the extracted image silhouettes from which the outline contour is extracted. Note the extraction errors which occur because of shadows and other fragments.

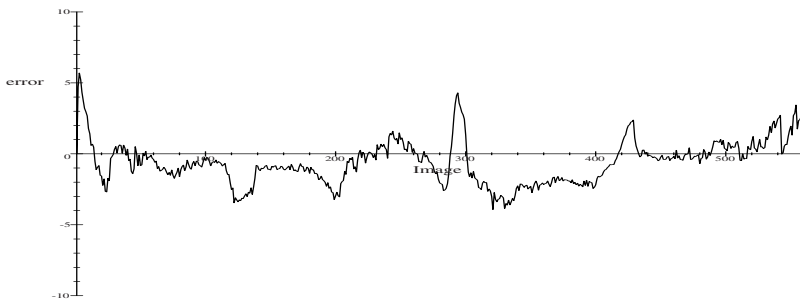


Fig. 11. The absolute error between the estimated angle and the ground truth in degrees. The maximum error is 5.73 degrees and the average error is 1.29 degrees.

error of the image sequence is 1.29 degrees. The errors are mainly dependent on the quality of image feature extraction, the calibration quality and the accuracy of the object model. Note that we are working with a full object model with changing aspects during the 360 degrees rotation.



Fig. 12. Example images of the tracked teapot. The hand grasping the teapot leads to outliers during the image silhouette extraction which are detected and eliminated during pose estimation.

4.2 Multiple Component Based Pose Estimation

We now present an extension of our approach for surface based free-form pose estimation to multiple surface patches. The reason is, that several objects can be represented through their including free-form parts more easily. Assume for example a teapot (see e.g. figure 12). It consists of a handle, a container and a spout.

We assume an extracted image silhouette and start with the reconstruction of the image contour points to 3D projection rays. This reconstruction is only estimated once for each image. Then the parts of the object model are projected in a virtual image. Since we assume the surface parts as rigidly coupled we extract and reconstruct **one** 3D silhouette of the surface model. Then we apply the 3D contour on our contour based pose estimation algorithm, which contains an ICP-algorithm and our gradient descent method for pose estimation. We then transform the surface model with the pose calculated from the contour based pose estimation algorithm and increase the low-pass approximation of each surface patch. Since the aspect of the object model can change after the iterated rigid transformations we generate a new 3D silhouette: The algorithm continues with a new projection of the object model in a virtual image and the loop repeats till the algorithm converges.

Figure 12 shows example images during an image sequence containing 350 images. This image sequence shows, that our algorithm is also able to deal with outliers during image processing which are caused by the human hand grasping the teapot.

4.3 Multiple-Components Mixed-Mode Pose Estimation

We now present experimental results where in addition to the silhouette also the internal object information is taken into account as discussed in section 3.3. The effect of using additional internal information is exemplarily shown in figure 13. As can be seen, the opening contour of the teapot is forced to the opening hole in the image and therefore stabilizes the result.

According to our previous experiment of the car on the turn table, we now present a similar experiment with the teapot. Furthermore, we estimate the

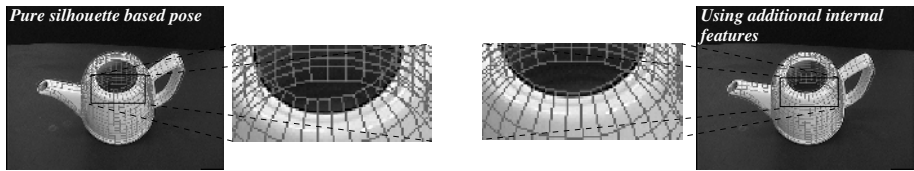


Fig. 13. Comparison of pose results of the pure silhouette based pose estimation algorithm (left) and the modified one (right) which uses additionally internal edge information.

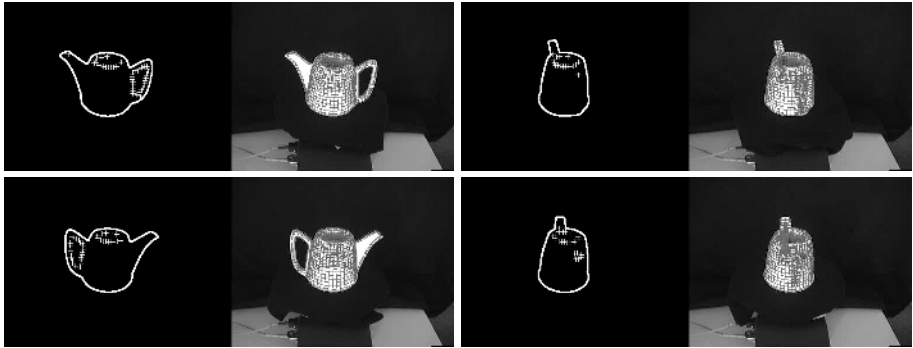


Fig. 14. Example images of the teapot on the turn table. The images to the left show the results after the image processing, the extracted contour and the used internal feature points. The images to the right show pose results.

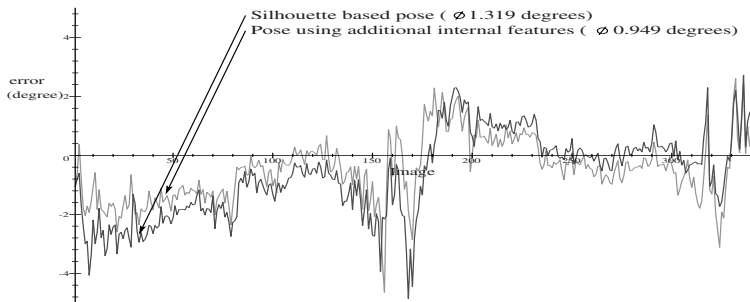


Fig. 15. The absolute pose error of the pure silhouette based pose estimation in comparison to the modified algorithm which uses additional internal object features. The error measure is the angle difference in degrees during the turntable image sequence.

absolute rotational error in degrees between the real pose and the ground truth of the teapot with and without using internal image information. Figure 14 shows example images of the image sequence and the image processing results, the used contour and the used internal image features. Figure 15 shows the comparison of the estimated pose with the angle of the turn table. The average error of the pure silhouette based pose estimation algorithm is 1.32 degrees and the average pose error by using additional internal features is 0.95 degrees. In this sequence we use

an image resolution of 384×288 pixels and the object is located approximately 1m in front of the camera. The calibration was performed with a calibration pattern containing 16 manually tracked reference points leading to a calibration error of 0.9 pixels for the reference points. This means we only work with coarsely calibrated cameras and low image resolution. The average computing time is 200 ms on a Linux 2GHz machine.

Indeed the comparison holds for just this scenario. For other objects the use of additional internal information might be useless or much more important than the extracted image silhouette. The aim of the experiments is to show that it is possible to extend the silhouette based pose estimation algorithm to scenarios which also use internal edge information of the surface model. To achieve this, we extend the surface model to a combination of free-form surface patches and free-form contour parts.

5 Discussion

In this work we present an extended approach for pose estimation of free-form surface models. Free-form surfaces are modeled by three 2D Fourier descriptors and low-pass information is used for approximation. The estimated 3D silhouette is then combined with the pose estimation constraints. Furthermore, an extension to the use of internal corner features of the object is presented. This leads to a combination of surface models with contour parts which is applied advantageously to the pose estimation problem. To deal with the basic problem of coupling projective geometry and kinematics we use a conformal geometric algebra. Though the equations are only symbolically explained, they present their simple geometric meaning within the chosen algebra. We further present experiments on different image sequences which visualize the properties of our algorithms e.g. in the context of noisy image data. The experiments show the stability of our algorithms with respect to noise and their capacity to deal with aspect changes during image sequences. Since we need up to 200ms per frame, our algorithms are fast and in the area of real-time.

Acknowledgments. This work has been supported by DFG Graduiertenkolleg No. 357, the EC Grant IST-2001-3422 (VISATEC) and by the DFG project RO 2497/1-1.

References

1. Arbter K. and Burkhardt H. Ein Fourier-Verfahren zur Bestimmung von Merkmalen und Schätzung der Lageparameter ebener Raumkurven. *Informationstechnik*, Vol. 33, No. 1, pp. 19–26, 1991.
2. Besl P.J. The free-form surface matching problem. *Machine Vision for Three-Dimensional Scenes*, Freemann H. (Ed.), pp. 25–71, Academic Press, 1990.
3. Bregler C. and Malik J. Tracking people with twists and exponential maps. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, pp. 8–15 1998.

4. Campbell R.J. and Flynn P.J. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding (CVIU)*, No. 81, pp. 166–210, 2001.
5. Drummond T. and Cipolla R. Real-time tracking of multiple articulated structures in multiple views. In *6th European Conference on Computer Vision, ECCV 2000, Dublin, Ireland, Part II*, pp. 20–36, 2000.
6. Granlund G. Fourier preprocessing for hand print character recognition. *IEEE Transactions on Computers*, Vol. 21, pp. 195–201, 1972.
7. Grimson W. E. L. Object Recognition by Computer. *The MIT Press, Cambridge, MA*, 1990.
8. Lowe D.G. Solving for the parameters of object models from image descriptions. In *Proc. ARPA Image Understanding Workshop*, pp. 121–127, 1980.
9. Li H., Hestenes D. and Rockwood A. Generalized homogeneous coordinates for computational geometry. In [14], pp. 27–52, 2001.
10. Murray R.M., Li Z. and Sastry S.S. A Mathematical Introduction to Robotic Manipulation. *CRC Press*, 1994.
11. Needham T. Visual Complex Analysis. *Oxford University Press*, 1997
12. Rosenhahn B. Pose Estimation Revisited. (PhD-Thesis) *Technical Report 0308, Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik*, 2003. Available at www.ks.informatik.uni-kiel.de
13. Rosenhahn B., Perwass C. and Sommer G. Pose estimation of free-form surface models. In *Pattern Recognition, 25th DAGM Symposium*, B. Michaelis and G. Krell (Eds.), Springer-Verlag, Berlin Heidelberg, LNCS 2781, pp. 574–581.
14. Sommer G., editor. Geometric Computing with Clifford Algebra. *Springer Verlag*, 2001.
15. Zang Z. Iterative point matching for registration of free-form curves and surfaces. *IJCV: International Journal of Computer Vision*, Vol. 13, No. 2, pp. 119–152, 1999.
16. Zerroug, M. and Nevatia, R. Pose estimation of multi-part curved objects. *Image Understanding Workshop (IUW)*, pp. 831–835, 1996

Interactive Image Segmentation Using an Adaptive GMMRF Model

A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr

Microsoft Research Cambridge UK,
7 JJ Thomson Avenue, Cambridge CB3 0FB, UK.
<http://www.research.microsoft.com/vision/cambridge>

Abstract. The problem of interactive foreground/background segmentation in still images is of great practical importance in image editing. The state of the art in interactive segmentation is probably represented by the graph cut algorithm of Boykov and Jolly (ICCV 2001). Its underlying model uses both colour and contrast information, together with a strong prior for region coherence. Estimation is performed by solving a graph cut problem for which very efficient algorithms have recently been developed. However the model depends on parameters which must be set by hand and the aim of this work is for those constants to be learned from image data.

First, a generative, probabilistic formulation of the model is set out in terms of a “Gaussian Mixture Markov Random Field” (GMMRF). Secondly, a pseudolikelihood algorithm is derived which jointly learns the colour mixture and coherence parameters for foreground and background respectively. Error rates for GMMRF segmentation are calculated throughout using a new image database, available on the web, with ground truth provided by a human segmenter. The graph cut algorithm, using the learned parameters, generates good object-segmentations with little interaction. However, pseudolikelihood learning proves to be frail, which limits the complexity of usable models, and hence also the achievable error rate.

1 Introduction

The problem of interactive image segmentation is studied here in the framework used recently by others [1,2,3] in which the aim is to separate, with minimal user interaction, a foreground object from its background so that, for practical purposes, it is available for pasting into a new context. Some studies [1,2] focus on inference of transparency in order to deal with mixed pixels and transparent textures such as hair. Other studies [4, 3] concentrate on capturing the tendency for images of solid objects to be coherent, via Markov Random Field priors such as the Ising model. In this setting, the segmentation is “hard” — transparency is disallowed. Foreground estimates under such models can be obtained in a precise way by graph cut, and this can now be performed very efficiently [5]. This has application to extracting the foreground object intact, even in camouflage — when background and foreground colour distributions overlap at least in part. We have not come across studies claiming to deal with transparency and camouflage simultaneously, and in our experience this is a very difficult combination. This paper therefore addresses the problem of hard segmentation problem in camouflage, and does not deal with the transparency issue.

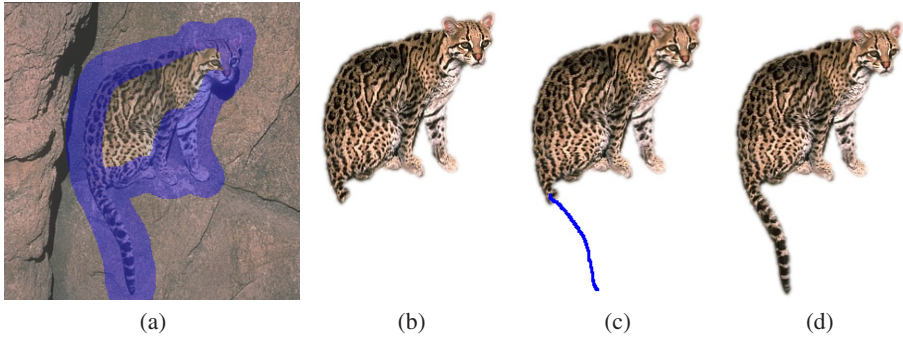


Fig. 1. Illustrating the GMMRF algorithm for interactive segmentation. The user draws a fat pen trail enclosing the object boundary (a), marked in blue. This defines the “trimap” with foreground/background/unclassified labels. The GMMRF algorithm produces (b). Missing parts of the object can be added efficiently: the user roughly applies a foreground brush (c), marked in blue, and the GMMRF method adds the whole region (d).

The interactive segmentation problem. The operation of adaptive segmentation by our model, termed a “Gaussian Mixture Markov Random Field” (GMMRF), is illustrated by the following example. The desired object has been cleanly separated from its background with a modest amount of user interaction (figure 1). Stripping away details of user interaction, the basic problem input consists of an image and its “trimap” as in figure 1a). The trimap defines training regions for foreground and background, and the segmentation algorithm is then applied to the “unclassified” region as shown, in which all pixels must be classified foreground or background as in fig. 1b). The classification procedure needs to apply:

- jointly across those pixels;
- matching the labels of adjoining labelled pixels;
- using models for colour and texture properties of foreground and background pixels learned from the respective training regions;
- benefiting from any general notions of region coherence that can be encoded in prior distributions.

2 Generative Models for Interactive Segmentation

This section reviews generative models for two-layer (foreground/background) colour images in order to arrive at the simplest capable, probabilistic model for interactive segmentation — the *contrast-sensitive GMMRF*. A generative, probabilistic model for image pixel data can be expressed in terms of colour pixel data \mathbf{z} , opacity variables α , opacity prior and data likelihood as follows.

Image. \mathbf{z} is an array of colour (RGB) indexed by the (single) index n :

$$\mathbf{z} = (z_1, \dots, z_n, \dots, z_N),$$

with corresponding hidden variables for transparency $\alpha = (\alpha_1, \dots, \alpha_N)$, and hidden variables for mixture index $\mathbf{k} = (k_1, \dots, k_N)$. Each pixel is considered, in general,

to have been generated as an additive combination of a proportion α_n ($0 \leq \alpha_n \leq 1$) of foreground colour with a proportion $1 - \alpha_n$ of background colour [2,1]. Here we concentrate attention on the hard segmentation problem in which $\alpha_n = 0, 1$ — binary valued.

Gibbs energy formulation. Now the posterior for α is given by

$$p(\alpha | \mathbf{z}) = \sum_{\mathbf{k}} p(\alpha, \mathbf{k} | \mathbf{z}) \quad (1)$$

and

$$p(\alpha, \mathbf{k} | \mathbf{z}) = \frac{1}{p(\mathbf{z})} p(\mathbf{z} | \alpha, \mathbf{k}) p(\alpha, \mathbf{k}). \quad (2)$$

This can be written as a Gibbs distribution, omitting the normalising constant $1/p(\mathbf{z})$ which is anyway constant with respect to α :

$$p(\alpha, \mathbf{k} | \mathbf{z}) \propto \frac{1}{Z_L} \exp -\mathcal{E} \text{ with } \mathcal{E} = L + U + V. \quad (3)$$

The intrinsic energies U and V encode the prior distributions:

$$p(\alpha) \propto \exp -U \text{ and } p(\mathbf{k} | \alpha) \propto \exp -V \quad (4)$$

and the extrinsic energy L defines the likelihood term

$$p(\mathbf{z} | \alpha, \mathbf{k}) = \prod_n p(z_n | \alpha_n, k_n) = \frac{1}{Z_L} \exp -L. \quad (5)$$

Simple colour mixture observation likelihood. A simple approach to modelling colour observations is as follows. At each pixel, foreground colour is considered to be generated randomly from one of K Gaussian mixture components with mean $\mu(\dots)$ and covariance $P(\dots)$ from the foreground, and likewise for the background:

$$p(z_n | \alpha_n, k_n) = \mathcal{N}(z; \mu(k_n, \alpha_n), P(k, \alpha_n)), \quad \alpha_n = 0, 1, \quad k_n = 1, \dots, K \quad (6)$$

and the components have prior probabilities

$$p(\mathbf{k} | \alpha) = \prod_n p(k_n | \alpha_n) \text{ with } p(k_n | \alpha_n) = \pi(k_n, \alpha_n). \quad (7)$$

The exponential coefficient for each component, referred to here as the “extrinsic energy coefficient” is denoted $S(k, \alpha) = \frac{1}{2}(P(k, \alpha))^{-1}$. The corresponding extrinsic term in the Gibbs energy is given by

$$L = \sum_n D_n \quad (8)$$

where

$$D_n = [z_n - \mu(k_n, \alpha_n)]^\top S(k_n, \alpha_n) [z_n - \mu(k_n, \alpha_n)]. \quad (9)$$

A useful special case is the *isotropic* mixture in which $S(k_n, \alpha_n) = \lambda(k_n, \alpha_n)I$.

Note that, for a pixelwise-independent likelihood model as in (6), the *partition function* Z_L for the likelihood decomposes multiplicatively across sites n . Since also the partition function Z_0 for the prior is always independent of α , the MAP estimate of α can be obtained (3) by minimising $\mathcal{E} - \log Z_L$ with respect to α and this can be achieved exactly, given that α_n is binary valued here, using a “minimum cut” algorithm [4] which has recently been developed [3] to achieve good segmentation in interactive time (around 1 second for a 500^2 image).

Simple opacity prior (No spatial interaction). The simplest choice of a joint prior $p(\alpha)$ is the spatially trivial one, decomposing into a product of marginals

$$p(\alpha) = \prod_n p(\alpha_n)$$

with, for example, $p(\alpha_n)$ uniform over $\alpha_n \in \{0, 1\}$ (hard opacity) or $\alpha_n \in [0, 1]$ (variable transparency). This is well known [3] to give poor results and this will be quantified in section 5.

Ising Prior. In order to convey a prior on object coherence, $p(\alpha)$ can be spatially correlated, for example via a first order Gauss-Markov interaction [6,7] as, for example, in the *Ising* prior:

$$p(\alpha) \propto \exp -U \text{ with } U = \gamma \sum_{m,n \in \mathcal{C}} [\alpha_n \neq \alpha_m], \quad (10)$$

where $[\phi]$ denotes the indicator function taking values 0, 1 for a predicate ϕ , and where \mathcal{C} is the set of all cliques in the Markov network, assumed to be two-pixel cliques here. The constant γ is the *Ising parameter*, determining the strength of spatial interaction. The MAP estimate of α can be obtained by minimising with respect to α the Gibbs energy (3) with L as before, and the Ising U (10). This can be achieved exactly, given that α_n is binary valued here, using a “minimum cut” algorithm [4]. In practice [3] the homogeneous MRF succeeds in enforcing some coherence, as intended, but introduces “Manhattan” artefacts that point to the need for a more subtle form of prior and/or data likelihood, and again this is quantified in section 5.

3 Incorporating Contrast Dependence

The Ising prior on opacity, being homogeneous, is a rather blunt instrument, and it is convincingly argued [3] that a “prior” that encourages object coherence *only* where contrast is low, is far more effective. However, a “prior” that is dependent on data in this way (dependent on data in that image gradients are computed from intensities \mathbf{z}) is not strictly a prior. Here we encapsulate the spirit of a contrast-sensitive “prior” more precisely as a gradient dependent likelihood term of the form

$$f(\nabla \mathbf{z} | \alpha, \beta, \gamma) \quad (11)$$

where β is a further coherence parameter in a Gauss-Markov process over z .

It turns out that the contrast term introduces a new technical issue in the data-likelihood model: long-range interactions are induced that fall outside the Markov framework, and therefore strictly fall outside the scope of graph cut optimisation. The long-range interaction is manifested in the partition function Z_L for the data likelihood. This is an inconvenient but inescapable consequence of the probabilistic view of the contrast-sensitive model. While it imperils the graph cut computation of the MAP estimate, and this will be addressed in due course, correct treatment of the partition function turns out to be critical for *adaptivity*. This is because of the well-known role of partition functions [7] in *parameter learning* for Gibbs distributions. Note also that recent advances in *Discriminative Random Fields* [8] which can often circumvent such issues, turn out not to be applicable to using the GMMRF model with trimap labelling.

3.1 Gibbs Energy for Contrast-Sensitive GMMRF

A modified Gibbs energy that takes contrast into account is obtained by replacing the term U in (10) by

$$U^+ = \gamma \sum_{m,n \in \mathcal{C}} [\alpha_n \neq \alpha_m] \exp -\frac{\beta}{\gamma} \|z_m - z_n\|^2, \quad (12)$$

which relaxes the tendency to coherence where image contrast is strong. The constant β is supposed to relate to γ via observation noise and we set

$$\frac{\gamma}{\beta} = C \langle \|z_m - z_n\|^2 \rangle, \text{ with } C = 4 \quad (13)$$

where $\langle \dots \rangle$ denotes expectation over the test image sample, and taking the constant $C = 4$ is justified later.

The results of minimising $\mathcal{E} = L + U^+$ (neglecting $\log Z_L$ — see later) gives considerably improved segmentation performance [3]. In our experience “Manhattan” artefacts are suppressed by the reduced tendency of segmentation boundaries to follow Manhattan geodesics, in favour of following lines of high contrast. Results in section 5 will confirm quantitatively the performance gain.

3.2 Probabilistic Model

In the contrast-sensitive version of the Gibbs energy, the term U^+ no longer corresponds to a prior for α , as it did in the homogeneous case (10). The entire Gibbs energy is now

$$\mathcal{E} = \sum_n D_n + \gamma \sum_{m,n \in \mathcal{C}} [\alpha_n \neq \alpha_m] \exp -\frac{\beta}{\gamma} \|z_m - z_n\|^2. \quad (14)$$

Adding a “constant” term

$$\gamma \sum_{m,n \in \mathcal{C}} (1 - \exp -\frac{\beta}{\gamma} \|z_m - z_n\|^2) \quad (15)$$

to \mathcal{E} has no effect on the minimum of $\mathcal{E}(\alpha)$ w.r.t. α , and transforms the problem in a helpful way as we will see. The addition of (15) gives $\mathcal{E} = U + L$ where U is the earlier Ising prior (10) and now L is given by

$$L = \sum_n D_n + \gamma \sum_{m,n \in \mathcal{C}} [\alpha_n = \alpha_m] (1 - \exp - \frac{\beta_{\alpha_n}}{\gamma} \|z_m - z_n\|^2), \quad (16)$$

with separate texture constants β_0, β_1 for foreground and background respectively. This is a fully generative, probabilistic account of the contrast-sensitive model, cleanly separating prior and likelihood terms. Transforming \mathcal{E} by the addition of the constant term (15) has had several beneficial effects. First it separates the energy into a component U which is active only at foreground/background region boundaries, and a component L whose contrast term acts only within region. It is thus clear that the parameter β is a textural parameter — and that is why it makes sense to learn separate parameters β_0, β_1 . Secondly when, for tractability in learning, the Gibbs energy is approximated by a quadratic energy in the next section 4, the transformation is in fact essential for the resulting data-likelihood MRF to be *proper* (ie capable of normalisation).

Inference of foreground and background labels from posterior. Given that L is now dependent on α and \mathbf{z} simultaneously, the partition function Z_L for the likelihood, which was a locally factorised function for the simple likelihood model of section 2, now contains non-local interactions over α . It is no longer strictly correct that the posterior can be maximised simply by minimising $\mathcal{E} = L + U$. Neglecting the partition function Z_L in this way can be justified, it turns out, by a combination of experiment and theory — see section 6.

MAP inference of α is therefore done by applying min cut as in [3] to the Gibbs energy \mathcal{E} . For this step we can either marginalise over \mathbf{k} , or maximize with respect to \mathbf{k} , the latter being computationally cheaper and tending to produce very similar results in practice.

4 Learning Parameters

This section addresses the critical issue of how mixture parameters $\mu(k, \alpha)$, $P(k, \alpha)$ and $\pi(k, \alpha)$ can be learned from data simultaneously with coherence parameters $\{\beta_\alpha\}$. In the simple uncoupled model with $\beta_\alpha = 0$ for $\alpha = 0, 1$ mixture parameters can be learned by conventional methods, but when coherence parameters are switched on, learning of all parameters is coupled non-trivially.

4.1 Quadratic Approximation

It would appear that the exponential form of L in (16) is an obstacle to tractability of parameter learning, and so the question arises whether it is an essential component of the model. Intuitively it seems well chosen because of the switching behaviour built into the exponential, that switches the model in and out of its “coherent” mode. Nonetheless,

in the interests of tractability we use a quadratic approximation, solely for the parameter learning procedure. The approximated form of the extrinsic energy (16) becomes:

$$L^* = \sum_n D_n + \sum_{m,n \in \mathcal{C}} \beta_{\alpha_n} [\alpha_n = \alpha_m] \|z_m - z_n\|^2. \quad (17)$$

and the approximation is good provided $\beta_{\alpha_n} \|z_m - z_n\|^2 < \gamma$.

Learning γ . Note that since the labelled data consists typically of separate sets of foreground and background pixels respectively (fig 1a,b) there is no training data containing the boundary between foreground and background. There is therefore no data over which the Ising term (10) can be observed, since γ no longer appears in the approximated L^* . Therefore γ cannot simply be learned from training data in this version of the interactive segmentation problem. However for the switching of the exponential term in (16) to act correctly it is clear that we must have

$$\exp - \frac{\beta_{\alpha_n}}{\gamma} \|z_m - z_n\|^2 \approx 1 \quad (18)$$

throughout regions of homogeneous texture so, over background for instance, we must have $\gamma \geq \beta_{\alpha_n} \|z_m - z_n\|^2$, which is also the condition for good quadratic approximation above. Given that the statistics of $z_m - z_n$ are found to be consistently Gaussian in practice, this is secured by (13).

4.2 Pseudolikelihood

A well established technique for parameter estimation in formally intractable MRFs like this one is to introduce a “pseudolikelihood function” [6] and maximise it with respect to its parameters. The pseudolikelihood function has the form

$$\mathcal{P} = \exp -\mathcal{E}^* \quad (19)$$

where \mathcal{E}^* will be called the “pseudo-energy”, and note that \mathcal{P} is free of any partition function. There is no claim that \mathcal{P} itself approximates the true likelihood, but that, under certain circumstances, its maximum coincides asymptotically with that of the likelihood [7] — asymptotically, that is, as the size of the data \mathbf{z} tends to infinity. Strictly the results apply for integer-valued MRFs, so formally we should consider \mathbf{z} to be integer-valued, and after all it does represent a set of quantised colour values.

Following [7], the pseudo-energy is defined to be

$$\mathcal{E}^* = \sum_n (-\log p(z_n | \mathbf{z}_n, \alpha, \mathbf{k})) \quad (20)$$

where $\mathbf{z}_n = \mathbf{z} \setminus \{z_n\}$ — the entire data array omitting z_n . Now

$$p(z_n | \mathbf{z}_n, \alpha, \mathbf{k}) = p(z_n | \{z_m, m \in \mathcal{B}_n\}, \alpha, \mathbf{k}) \quad (21)$$

by the Markov property, where \mathcal{B}_n is the “Markov blanket” at grid point n — the set of its neighbours in the Markov model. For the earlier example of a first-order MRF, \mathcal{B}_n

simply contains the N, S, E, W neighbours of n . Taking into account the earlier details of the probability distribution over the Markov structure, it is straightforward to show that, over the training set

$$p(z_n \mid \{z_m, m \in \mathcal{B}_n\}, \alpha, \mathbf{k}) \propto \exp - \left[D_n + \sum_{m \in \mathcal{B}_n} V_{nm} \right] \quad (22)$$

where

$$V_{nm} = \beta_{\alpha_n} [\alpha_m = \alpha_n] \|z_n - z_m\|^2. \quad (23)$$

Terms $\gamma[\alpha_m \neq \alpha_n]$ from U have been omitted since they do not occur in the training sets of the type used here (fig 1), as mentioned before. Finally, the pseudo-likelihood energy function

$$\mathcal{E}^* = \sum_n \mathcal{E}_n^* \text{ with } \mathcal{E}_n^* = Z_n^* + D_n + \sum_{m \in \mathcal{B}_n} V_{nm} \quad (24)$$

and $\exp -Z_n^*$ is the (local) partition function.

4.3 Parameter Estimation by Autoregression over the Pseudolikelihood

It is well known that the parameters of a Gaussian MRF can be obtained from pseudolikelihood as an auto-regression estimate [9]. For tractability, we split the estimation problem up into $2K$ problems, one for each foreground and background mixture component, treated independently except for sharing common constants β_0 and β_1 . For this purpose, the mixture index k_n at each pixel is determined simply by maximisation of the local likelihood:

$$k_n = \arg \max_k p(z_n \mid \alpha_n, k) \pi_k^{\alpha_n}. \quad (25)$$

Further, for tractability, we restrict the data $\{z_n\}$ to those pixels (much the majority in practice) whose foreground/background label α agrees with all its neighbours — ie the n for which

$$\alpha_m = \alpha_n \text{ for all } m \in \mathcal{B}_n.$$

Observables z_n with a given class label α_n and component index k_n are then dealt with together, in accordance with the pseudolikelihood model above, as variables from the regression

$$z_n - \bar{z} \mid \mathbf{z}_n \sim \mathcal{N}(A(x_n - \bar{z}), P) \quad (26)$$

where

$$x_n = \frac{1}{M} \sum_{m \in \mathcal{B}_n} z_m, \quad (27)$$

and $M = |\mathcal{B}_n|$. The mean colour is estimated simply as

$$\bar{z} = \langle z \rangle \quad (28)$$

where $\langle \dots \rangle$ denotes the sample mean over pixels from class α and with component index k . We can solve for A and P using standard linear regression as follows:

$$A = \langle (z - \bar{z})(x - \bar{z})^\top \rangle (\langle (x - \bar{z})(x - \bar{z})^\top \rangle)^{-1} \quad (29)$$

and

$$P = \langle \epsilon \epsilon^\top \rangle \text{ where } \epsilon = z - \bar{z} - A(x - \bar{z}). \quad (30)$$

Lastly, model parameters for each colour component, for instance of the background, should be obtained to satisfy

$$M\beta_0 I = P^{-1}A \quad (31)$$

$$S(k_n, \alpha_n) = P^{-1} - M\beta_0 I, \quad (32)$$

$$\mu(k_n, \alpha_n) = \bar{z}, \quad (33)$$

and similarly for the foreground.

Note there are some technical problems here. First is that (31) represents a constraint that is not necessarily satisfied by P and A , and so the regression needs to be solved under this constraint. Second is that in (32) $S(k_n, \alpha_n)$ should be positive definite but this constraint will not automatically be obeyed by the solution of the autoregression. Thirdly that one value of β_0 needs to satisfy the set of equations above for all background components. The first problem is dealt with by restricting S to be isotropic so that P and A must also be isotropic and the entire set of equations are solved straightforwardly under isotropy constraints. (In other words, each colour component is regressed independently of the others.) It turns out that this also solves the second problem. An unconstrained autoregression on typical natural image data, with general symmetric matrices for the $S(k_n, \alpha_n)$, will, in our experience, often lead to a non-positive definite solution for $S(k_n, \alpha_n)$ and this is unusable in the model. Curiously this problem with pseudolikelihood and autoregression seems not to be generally acknowledged in standard texts [7, 9]. Empirically however we have found that the problem ceases with isotropic $S(k_n, \alpha_n)$, and so we have used this throughout our experiments. The use of isotropic components seems not to have much effect on quality provided that, of course, a larger number of mixture components must be used than for equivalent performance with general symmetric component-matrices. Lastly, the tying of β_0 across components can be achieved simply by averaging *post-hoc*, or by applying the tying constraint explicitly as part of the regression which is, it turns out, quite tractable (details omitted).

5 Results

Testing of the GMMRF segmentation algorithms uses a database of 50 images. We compare the performance of: i) Gaussian colour models, with no spatial interaction; ii) the simple Ising model; iii) the full contrast-sensitive GMMRF model with fixed interaction parameter γ ; iv) the full GMMRF with learned parameters. Results are obtained using 4-connectivity, and isotropic Gaussian mixtures with $K = 30$ components as the data potentials D_n .

Test Database. The database contains 15 training and 35 test images, available online¹. The database contrasts with the form of ground truth supplied with the Berkeley

¹ <http://www.research.microsoft.com/vision/cambridge/segmentation/>

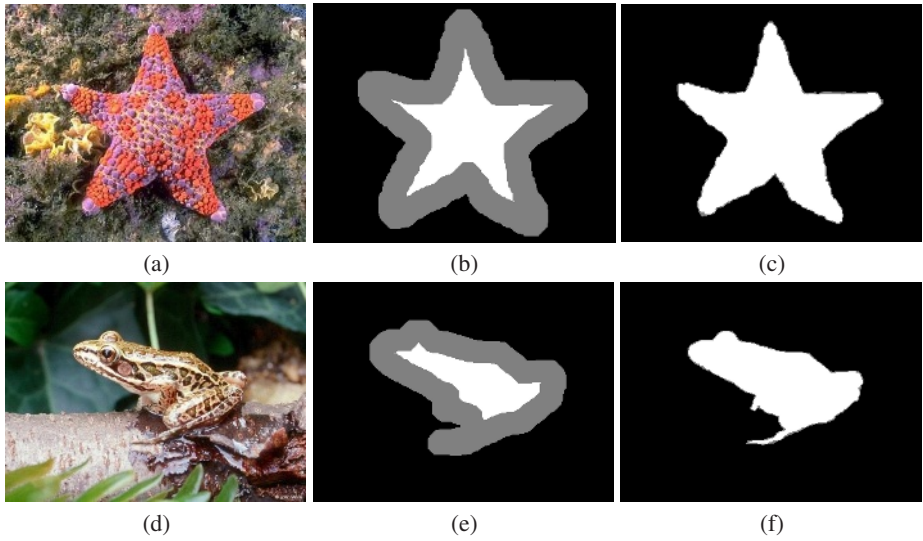


Fig. 2. (a,d) Two images from the test database. (b,e) User defined trimap with foreground (white), background (black) and unclassified (grey). (c,f) Expert trimap which classifies pixels into foreground (white), background (black) and unknown (grey); unknown here refers to pixels too close to the object boundary for the expert to classify, including mixed pixels.

database² which is designed to test exhaustive, for bottom up segmentation [10]. Each image in our database contains a foreground object in a natural background environment (see fig. 2). Since the purpose of the dataset is to evaluate various algorithms for *hard* image segmentation, objects with no or little transparency are used. Consequently, partly transparent objects like trees, hair or glass are not included. Two kinds of labeled trimaps are assigned to each image. The first is the user trimap as in fig. 2(b,e). The second is an “expert trimap” obtained from painstaking tracing of object outlines with a fine pen (fig. 2(c,f)). The fine pen-trail covers possibly mixed pixels on the object boundary. These pixels are excluded from the error rate measures reported below, since there is no definitive ground truth as to whether they are foreground or background.

Evaluation. Segmentation error rate is defined as

$$\epsilon = \frac{\text{no. misclassified pixels}}{\text{no. pixels in unclassified region}}, \quad (34)$$

where “misclassified pixels” excludes those from the unclassified region of the expert trimap. This simple measurement is sufficient for a basic evaluation of segmentation performance. It might be desirable at some later date to devise a second measure that quantifies the degree of user effort that would be required to correct errors, for example by penalising scattered error pixels.

² <http://www.cs.berkeley.edu/projects/vision/grouping/segbench/>

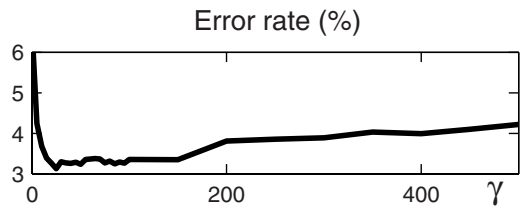


Fig. 3. The error rate on the training set of 15 images, using different values for γ . The minimum error is achieved for $\gamma = 25$. The GMMRF model uses isotropic Gaussians and 4-connectivity.

Segmentation model	Error rate
GMMRF; discriminatively learned $\gamma = 20$ ($K = 10$ full Gaussian)	7.9%
Learned GMMRF parameters ($K = 30$ isotropic Gauss.)	8.6%
GMMRF; discriminatively learned $\gamma = 25$ ($K = 30$ isotropic Gaussian)	9.4%
Strong interaction model ($\gamma = 1000$; $K = 30$ isotropic Gaussian)	11.0%
Ising model ($\gamma = 25$; $K = 30$ isotropic Gaussian)	11.2%
Simple mixture model – no interaction ($K = 30$ isotropic Gaussian)	16.3%

Fig. 4. Error rates on the test data set for different models and parameter determination regimes. For isotropic Gaussians, the full GMMRF model with learned parameters outperforms both the full model with discriminatively learned parameters, and simpler alternative models. However, exploiting a full Gaussian mixture model improves results further.

Test database scores. In order to compare the GMMRF method with alternative methods, a fixed value of the Ising parameter γ is learned discriminatively by optimising performance over the training set (fig. 3), giving a value of $\gamma = 25$. Then the accompanying β constant is fixed as in (13). For full Gaussians and 8-connectivity the learned value was $\gamma = 20$. The performance on the test set is summarized in figure 4 for the various different models and learning procedures. Results for the two images of figure 2 from the test database, are shown in figure 5. As might be expected, models with very strong spatial interaction, simple Ising interaction, or without any spatial interaction at all, all perform poorly. The model with no spatial interaction has a tendency to generate many isolated, segmented regions (see fig. 5). A strong interaction model ($\gamma = 1000$) has the effect of shrinking the the object with respect to the true segmentation. The Ising model, with $\gamma = 25$ set by hand, gives slightly better results, but introduces “Manhattan” artefacts — the border of the segmentation often fails to correspond to image edges. The inferiority of the Ising model and the “no interaction model” has been demonstrated previously [3] but here is quantified for the first time.

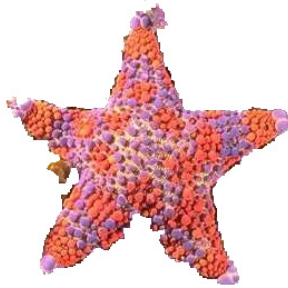
In contrast, the GMMRF model with learned γ is clearly superior. For isotropic Gaussians and 4-connectivity the GMMRF model with parameters *learned* by the new pseudolikelihood algorithm leads to slightly better results than using the discriminatively learned γ .

The lowest error rate, however, was achieved using full covariance Gaussians in the GMMRF with discriminatively learned γ . We were unable to compare with pseudolikelihood learning; the potential instability of pseudolikelihood learning (section 4) turns out to be an overwhelming obstacle when using full covariance Gaussians.

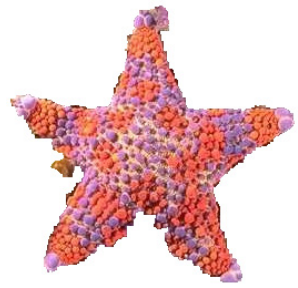
GMMRF; $\gamma = 20$; full Gauss.
error = 1.5%



GMMRF; γ learned
error = 4.5%



GMMRF; $\gamma = 25$
error = 4.7%



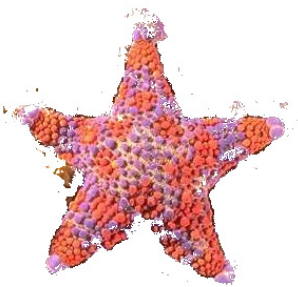
Ising model; $\gamma = 25$
error = 5.9%



Strong interaction; $\gamma = 1000$
error = 7.6%



No interaction
error = 8.9%



GMMRF; $\gamma = 20$; full Gauss.
error = 6.8%



GMMRF; γ learned
error = 7.0%



GMMRF; $\gamma = 25$
error = 9.7%



Ising model; $\gamma = 25$
error = 10.7%



Strong interaction; $\gamma = 1000$
error = 15.9%



No interaction
error = 20.0%



Fig. 5. Results for various segmentation algorithms (see fig. 4) for the two images shown in fig. 2. For both examples, the error rate increases from top left to bottom right. GMMRF with pseudolikelihood learning outperforms the GMMRF with discriminatively learned γ parameters, and the various simpler alternative models. The best result is achieved however using full covariance Gaussians and discriminatively learned γ .

6 Discussion

We have formalised the energy minimization model of Boykov and Jolly [3] for foreground/background segmentation as a probabilistic GMMRF and developed a pseudolikelihood algorithm for parameter learning. A labelled database has been constructed for this task and evaluations have corroborated and quantified the value of spatial interaction models — the Ising prior and the contrast-sensitive GMMRF. Further, evaluation has shown that parameter learning for the GMMRF by pseudolikelihood is effective. Indeed, it is a little more effective than simple discriminative learning for a comparable model (isotropic GMMRF); but the frailty of pseudolikelihood learning limits the complexity of model that can be used (eg full covariance GMMRF is impractical) and that in turn limits achievable performance. A number of issues remain for discussion, as follows.

DRF. The Discriminative Random Field model [8] has recently been shown to be very effective for image classification tasks. It has the great virtue of banishing the issues concerning the likelihood partition function Z_L that affect the GMMRF. However it can be shown that the DRF formulation cannot be used with the form of GMMRF developed here, and trimap labelled data, because the parameter learning algorithm breaks down (details omitted for lack of space).

Line process. The contrast-sensitive GMMRF has some similarity to the well known line process model [11]. In fact it has an important additional feature, that the observation model is a non-trivial MRF with spatial interaction (the contrast term), and this is a crucial ingredient in the success of contrast-sensitive GMMRF segmentation.

Likelihood partition function. As mentioned in section 3.2, the partition function Z_L depends on α and this dependency should be taken into account when searching the MAP estimate of α . However, for the quadratically approximated extrinsic energy (17), this partition function is proportional to the determinant of a sparse precision matrix, which can be numerically computed for given parameters and α . Within the range of values used in practice for the different parameters, we found experimentally that

$$\log Z_L(\alpha) = \text{const} + \kappa \sum_{m,n \in \mathcal{C}} [\alpha_n \neq \alpha_m], \quad (35)$$

with κ varying within a range $(0, 0.5)$. Hence, by ignoring $\log Z_L$ in the global energy to be minimised, we assume implicitly that the prior is effectively Ising with slightly weaker interaction parameter $\gamma - \kappa$. Since we have seen that graph cut is relatively insensitive (fig. 3) to perturbations in γ , this justifies neglect of the extrinsic partition fn Z_L and the application of graph cut to the Gibbs energy alone.

Adding parameters to the MRF. It might seem that segmentation performance could be improved further by allowing more general MRF models. They would have greater numbers of parameters, more than could reasonably be set by hand, and this ought

to press home the advantage of the new parameter learning capability. We have run preliminary experiments with i) spatially anisotropic clique potentials, ii) larger neighbourhoods (8-connected) and iii) independent, unpooled foreground and background texture parameters β_0 and β_1 (using min cut over a directed graph). However, in all cases error rates were substantially worsened. A detailed analysis of these issues is part of future research.

Acknowledgements. We gratefully acknowledge discussions with and assistance from P. Anandan, C.M. Bishop, B. Frey, T. Werner, A.L. Yuille and A. Zisserman.

References

1. Chuang, Y.Y., Curless, B., Salesin, D., Szeliski, R.: A Bayesian approach to digital matting. In: Proc. Conf. Computer Vision and Pattern Recognition. (2001) CD-ROM
2. Ruzon, M., Tomasi, C.: Alpha estimation in natural images. In: Proc. Conf. Computer Vision and Pattern Recognition. (2000) 18–25
3. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: Proc. Int. Conf. on Computer Vision. (2001) CD-ROM
4. Greig, D., Porteous, B., Seheult, A.: Exact MAP estimation for binary images. *J. Royal Statistical Society* **51** (1989) 271–279
5. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? *IEEE Trans. on Pattern Analysis and Machine Intelligence* **in press** (2003)
6. Besag, J.: On the statistical analysis of dirty pictures. *J. Roy. Stat. Soc. Lond. B.* **48** (1986) 259–302
7. Winkler, G.: Image analysis, random fields and dynamic Monte Carlo methods. Springer (1995)
8. Kumar, S., Hebert, M.: Discriminative random fields: A discriminative framework for contextual interaction in classification. In: Proc. Int. Conf. on Computer Vision. (2003) CD-ROM
9. Descombes, X., Sigelle, M., Preteux, F.: GMRf parameter estimation in a non-stationary framework by a renormalization technique. *IEEE Trans. Image Processing* **8** (1999) 490–503
10. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. *Int. J. Computer Vision* **43** (2001) 7–27
11. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **6** (1984) 721–741

Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model

Xianghua Ying and Zhanyi Hu*

National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, 100080 P.R.China

{xhying, huzy}@nlpr.ia.ac.cn,
<http://nlpr-web.ia.ac.cn/English/rv/~ying/>

Abstract. There are two kinds of omnidirectional cameras often used in computer vision: central catadioptric cameras and fisheye cameras. Previous literatures use different imaging models to describe them separately. A unified imaging model is however presented in this paper. The unified model in this paper can be considered as an extension of the unified imaging model for central catadioptric cameras proposed by Geyer and Daniilidis. We show that our unified model can cover some existing models for fisheye cameras and fit well for many actual fisheye cameras used in previous literatures. Under our unified model, central catadioptric cameras and fisheye cameras can be classified by the model's characteristic parameter, and a fisheye image can be transformed into a central catadioptric one, vice versa. An important merit of our new unified model is that existing calibration methods for central catadioptric cameras can be directly applied to fisheye cameras. Furthermore, the metric calibration from single fisheye image only using projections of lines becomes possible via our unified model but the existing methods for fisheye cameras in the literatures till now are all non-metric under the same conditions. Experimental results of calibration from some central catadioptric and fisheye images confirm the validity and usefulness of our new unified model.

1 Introduction

In many computer vision applications, including robot navigation, 3D reconstruction, and image-based rendering, a camera with a quite large field of view (FOV) is required. A conventional camera has a very limited field of view, therefore some omnidirectional cameras, such as cameras with fisheye lenses, multi-camera systems and catadioptric imaging systems are employed. There are some representative implementations of omnidirectional cameras described in [17]. In

* This work was supported by the National Key Basic Research and Development Program China (973) under grant No. 2002CB312104, and the National Natural Science Foundation China under grant No. 60121302.



Fig. 1. (a) An image from a central catadioptric camera which is combined a perspective camera with a hyperboloidal mirror, designed by the Center for Machine Perception, Czech Technical University, and its field of view is 217 degrees. (b) An image from a fisheye camera which is a Nikon COOLPIX 990 with FC-E8 fisheye lenses, and its FOV is 183 degrees. The two images are taken in almost same position and direction.

this paper, we will present a unified imaging model for catadioptric and fisheye cameras with single viewpoint constraint. A catadioptric camera is an imaging device which combines a pinhole and a reflective mirror. Baker and Nayar [1] derive the complete class of single-lens single-mirror catadioptric sensors which have a single effective viewpoint. A catadioptric camera with a single viewpoint is called central catadioptric camera. A fisheye camera is an imaging device which mounts a fisheye lens on a conventional camera. As noted in [4], fisheye cameras do not have a single projection center but a locus of projection centers called diacaustic. However, in many computer vision applications, such as robot navigation, image-based rendering etc., it is reasonable to assume that the small projection locus can be approximated by a single viewpoint if calibration accuracy under this assumption can satisfy the requirement of applications. Most existing literatures [3,5,6,7,8,10,14,15,19,20,22,24] use this assumption as well as this paper. Images taken from a central catadioptric camera and a fisheye camera are shown in Fig. 1a and 1b respectively.

The motivations for proposing a unified imaging model for central catadioptric and fisheye cameras are based on the following observations: Similar to that under central catadioptric cameras lines in space are projected into conics in the catadioptric image [18,21], we find that lines in space are also projected into conics in the fisheye image using Nikon FC-E8 fisheye lenses mounted on a Nikon COOLPIX 990. Smith et al. [20] claim that space lines are projected into conics using some fisheye camera if the fisheye camera satisfies a two-step model via a quadric surface. Bräuer-Burchardt and Voss [5] discover that the projections of space lines are circles under some fisheye cameras. Nene and Nayar [18] note that the projections of space lines are circles too under a central para-catadioptric camera. All these imply that there should exist a unified model for central catadioptric and fisheye cameras.

The unified imaging model for central catadioptric and fisheye cameras presented in this work is an extension of the unified imaging model for central catadioptric camera proposed by Geyer and Daniilidis [12]. We show that this new unified model can cover some existing models for fisheye camera [5,8,20] and fit well for many actual fisheye cameras used in [7,14,19,22,24]. The equivalence of different imaging models is rigorously proved. An important merit for proposing this new unified model is that calibration methods in [2,11,13,25] for central catadioptric cameras can be directly applied to fisheye cameras. Furthermore, the metric calibration from single fisheye image only using projections of lines becomes possible via our unified model whereas existing methods for fisheye cameras in [5,6,7,14,22] are all non-metric under the same conditions.

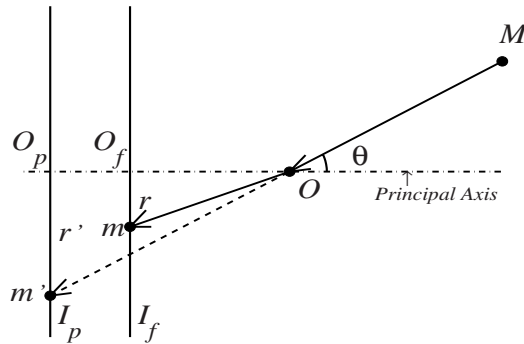


Fig. 2. Fisheye imaging process and its corresponding perspective projection. O is the projection center, I_p represents the perspective projection image plane, and I_f represents the fisheye image plane. For a space point M , its fisheye image is m and its perspective image is m' .

1.1 Related Work

There exists a unified imaging model for central catadioptric cameras proposed by Geyer and Daniilidis [12]. But for fisheye cameras, there exist many different imaging models in the literature. These existing models can be broadly divided into the following two categories:

1. Transformation between fisheye image and its corresponding perspective image

For a space point $\mathbf{M} = (X, Y, Z)^T$, let its fisheye image point $\mathbf{m} = (x, y)^T$, the corresponding perspective image point $\mathbf{m}' = (x', y')^T$, the origin of the world coordinate system located at the projection center, and the origins of the two image coordinate systems all located at the principal points (see Fig. 2). Therefore, we have $r = \sqrt{x^2 + y^2}$ and $r' = \sqrt{x'^2 + y'^2}$. The transformation between fisheye image and its corresponding perspective image can be represented by $(x, y) \xleftrightarrow{T} (x', y')$ or $r \xleftrightarrow{T} r'$. Basu et al. [3] present a logarithmic mapping

model, and Shah et al. [19] present a polynomial one. Recently, Bräuer-Burchardt et al. [5] and Fitzgibbon [8] propose a rational function model as:

$$r' = k_1 \frac{r}{1 - k_2 r^2}, \quad (1)$$

where k_1 and k_2 are two parameters of the model.

2. Transformation between fisheye image and its corresponding captured rays

The angle between a captured ray OM and the principle axis is denoted by θ (Fig. 2). The transformation between fisheye image and its corresponding captured rays is described as $(x, y) \xleftrightarrow{T} \left(\frac{X}{\sqrt{X^2+Y^2+Z^2}}, \frac{Y}{\sqrt{X^2+Y^2+Z^2}}, \frac{Z}{\sqrt{X^2+Y^2+Z^2}} \right)$ or $r \xleftrightarrow{T} \theta$. Xiong and Turkowski [24] present a polynomial model. Micusik and Pajdla [15] propose a rational function model. Miyamoto [16] uses equidistance projection model, and Fleck [10] employs stereographic projection model of the viewing sphere. Smith et al. [20] propose a two-step projection model via a quadric surface: the first step is that a 3D space point is projected to a point on the quadric surface which is the intersection of the captured ray with the quadric surface, and the second step is that the intersection point is orthographically projected to an image plane.

2 The Unified Imaging Model

In this section, we start with a generalized two-step projection model via a quadric surface, and then specify the generalized model in many different ways to obtain the unified imaging model and other two-step projection models. The proofs of the equivalence among these models are also provided in this section.

2.1 Generalized Two-Step Projection via a Quadric Surface (TSP0)

The generalized two-step projection is defined as follows: a space point is first projected to a point on the quadric surface, and then perspectively projected into a point on the image plane using a pinhole (see Fig. 3). There are some specific points about this generalized model:

1. The quadric surface is a revolution of a conic section about one of its principal axes where the foci lie. The effective viewpoint is located at one of the foci of the conic section and the pinhole lies on the revolution axis. Note that the pinhole can be located anywhere on the revolution axis. The image plane of the pinhole is perpendicular to the revolution axis. There are two different configurations with the different positions of the pinhole related to the quadric surface as shown in Fig. 3a and 3b.

2. The pinhole can be replaced by an orthographic camera. The quadric surface can be ellipsoidal ($0 < e < 1$, where e is the eccentricity of the conic section), paraboloidal ($e = 1$), hyperboloidal ($e > 1$) or some degenerated cases, such as, spherical ($e \rightarrow 0$) or planar ($e \rightarrow \infty$).

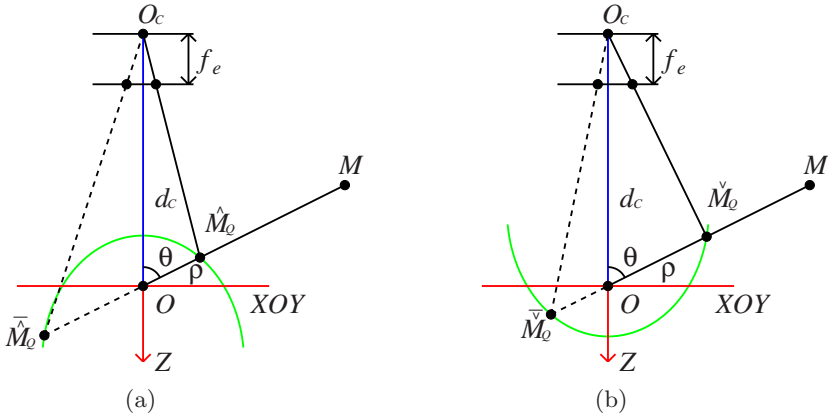


Fig. 3. A generalized two-step projection via a quadric surface. The effective viewpoint O is located at one of the foci of the quadric surface and the pinhole is located at O_C . A space point M is first projected into a point on the quadric surface, and then perspectively projected into a point on the image plane of the pinhole.

3. It is assumed that the quadric surface is transparent, i.e., the line OM intersects the quadric surface in two points. Note that the generalized two-step projection need not obey the law of reflection.

The world coordinate system $OXYZ$ is established as shown in Fig. 3. The origin of the coordinate system is located at the effective viewpoint O . The generalized projection of a space point $\mathbf{M} = (X, Y, Z)^T$ is denoted as:

$$(x, y) = Q_{e,p,f_e,d_c}(X, Y, Z), \quad (2)$$

where e is the eccentricity of the conic section, p is the distance from the focus to the directrix of the conic section, f_e is the effective focal length of the pinhole, and d_c is the distance from the origin to the pinhole. Because of the ambiguity caused by the transparency of the quadric surface and the position of the pinhole related to the quadric surface, there are four different projection points on the quadric surface: \hat{M}_Q , \check{M}_Q , \tilde{M}_Q and \bar{M}_Q (see Fig. 3a and 3b). Hence, there are four different kinds of projections:

$$Q_{e,p,f_e,d_c} = \hat{Q}_{e,p,f_e,d_c} \cup \check{Q}_{e,p,f_e,d_c} \cup \tilde{Q}_{e,p,f_e,d_c} \cup \bar{Q}_{e,p,f_e,d_c}.$$

Because

$$\left(\bar{Q}_{e,p,f_e,d_c} \cup \tilde{Q}_{e,p,f_e,d_c} \right) (X, Y, Z) = \left(\hat{Q}_{e,p,f_e,d_c} \cup \check{Q}_{e,p,f_e,d_c} \right) (-X, -Y, -Z), \quad (3)$$

we will first find out \hat{Q}_{e,p,f_e,d_c} and \check{Q}_{e,p,f_e,d_c} , and then derive \tilde{Q}_{e,p,f_e,d_c} and \bar{Q}_{e,p,f_e,d_c} using (3).

The ray OM intersects the quadric surface at \hat{M}_Q in Fig. 3a (or \check{M}_Q in Fig. 3b), the distance OM_Q (or OM_Q) satisfies:

$$\rho = \frac{ep}{1 \pm e \cos \theta}, \quad (4)$$

where

$$\cos \theta = -\frac{Z}{\sqrt{X^2 + Y^2 + Z^2}}. \quad (5)$$

\hat{M}_Q (or \check{M}_Q) satisfies:

$$(X_Q, Y_Q, Z_Q) = \frac{\rho}{\sqrt{X^2 + Y^2 + Z^2}}(X, Y, Z), \quad (6)$$

where (X_Q, Y_Q, Z_Q) represents the world coordinates of \hat{M}_Q (or \check{M}_Q). We assume the intrinsic matrix of the pinhole is:

$$\mathbf{K} = \begin{bmatrix} f_e & 0 & 0 \\ 0 & f_e & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (7)$$

Therefore, the projection of \hat{M}_Q (or \check{M}_Q) on the image plane satisfies:

$$\lambda \tilde{\mathbf{m}} = \lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_e & 0 & 0 \\ 0 & f_e & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_C \end{bmatrix} \begin{bmatrix} X_Q \\ Y_Q \\ Z_Q \\ 1 \end{bmatrix}, \quad (8)$$

where λ is an unknown scale factor, and $\tilde{\mathbf{m}}$ is the homogeneous coordinates of $\mathbf{m} = (x, y)^T$. By eliminating λ from (8) and with (6), (4) and (5), we obtain:

$$(x, y) = \left(\hat{Q}_{e,p,f_e,d_c} \cup \check{Q}_{e,p,f_e,d_c} \right) (X, Y, Z) = \left(\frac{epf_e X}{d_C \sqrt{X^2 + Y^2 + Z^2} + e(p \mp d_C)Z}, \frac{epf_e Y}{d_C \sqrt{X^2 + Y^2 + Z^2} + e(p \mp d_C)Z} \right). \quad (9)$$

From (3) and (9), we obtain:

$$(x, y) = \left(\hat{Q}_{e,p,f_e,d_c} \cup \check{Q}_{e,p,f_e,d_c} \right) (X, Y, Z) = \left(-\frac{epf_e X}{d_C \sqrt{X^2 + Y^2 + Z^2} - e(p \mp d_C)Z}, -\frac{epf_e Y}{d_C \sqrt{X^2 + Y^2 + Z^2} - e(p \mp d_C)Z} \right). \quad (10)$$

2.2 Image Formation of Central Catadioptric Camera (TSP1)

Baker and Nayar [1] show that the only useful physically realizable mirror surfaces of catadioptric cameras that produce a single viewpoint are planar, ellipsoidal, hyperboloidal, and paraboloidal. For a planar mirror, given a fixed

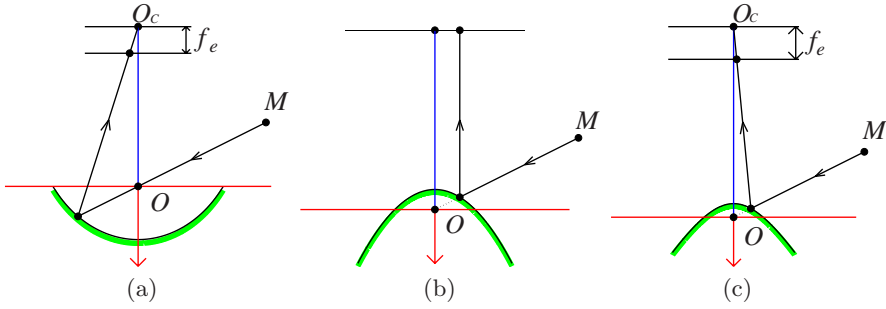


Fig. 4. Image formations of different kinds of central catadioptric cameras. (a) ellipsoidal, (b) paraboloidal, (c) hyperboloidal. Obviously, these models obey the law of reflection.

viewpoint and a pinhole, the configuration is the perpendicular bisector of the line joining the pinhole to the viewpoint. Under an orthographic camera, the only solution is a paraboloidal mirror with the effective viewpoint at the focus of the paraboloid. The hyperboloidal mirror satisfies the fixed viewpoint constraint when the pinhole and the viewpoint are located at the two foci of the hyperboloid. The ellipsoidal mirror can be configured in a similar way as the hyperboloidal one (see Fig. 4abc). Obviously, the four cases are all special cases of the generalized model. Since the planar catadioptric camera is equivalent to a pinhole, we do not discuss it here.

For the case of ellipsoid (see Fig. 4a), we have $0 < e < 1$ and $d_C = \frac{2e^2 p}{1-e^2}$ (the distance of the two foci of the ellipsoid), then we obtain:

$$\begin{aligned} (x, y) &= E_{e,p,f_e}(X, Y, Z) = \tilde{Q}_{e,p,f_e,d_c}(X, Y, Z) \\ &= \left(\frac{\frac{1-e^2}{1+e^2} f_e X}{-\frac{2e}{1+e^2} \sqrt{X^2 + Y^2 + Z^2} + Z}, \frac{\frac{1-e^2}{1+e^2} f_e Y}{-\frac{2e}{1+e^2} \sqrt{X^2 + Y^2 + Z^2} + Z} \right). \end{aligned} \quad (11)$$

For the case of paraboloid (see Fig. 4b), we have $e = 1$ and $f_e \rightarrow \infty, d_C \rightarrow \infty, \frac{f_e}{d_C} \rightarrow 1$. Since the paraboloidal has two ambiguous configurations, we obtain:

$$\begin{aligned} (x, y) &= P_p(X, Y, Z) = \left(\hat{Q}_{1,p,\infty,\infty} \cup \tilde{\hat{Q}}_{1,p,\infty,\infty} \right)(X, Y, Z) \\ &= \left(\mp \frac{pX}{-\sqrt{X^2 + Y^2 + Z^2} + Z}, \mp \frac{pY}{-\sqrt{X^2 + Y^2 + Z^2} + Z} \right). \end{aligned} \quad (12)$$

For the case of hyperboloid (see Fig. 4c), we have $e > 1$ and $d_C = \frac{2e^2 p}{1-e^2}$ (the distance of the two foci of the hyperboloid), then we obtain:

$$\begin{aligned} (x, y) &= H_{e,p,f_e}(X, Y, Z) = \hat{Q}_{e,p,f_e,d_c}(X, Y, Z) \\ &= \left(\frac{\frac{1-e^2}{1+e^2} f_e X}{-\frac{2e}{1+e^2} \sqrt{X^2 + Y^2 + Z^2} + Z}, \frac{\frac{1-e^2}{1+e^2} f_e Y}{-\frac{2e}{1+e^2} \sqrt{X^2 + Y^2 + Z^2} + Z} \right). \end{aligned} \quad (13)$$

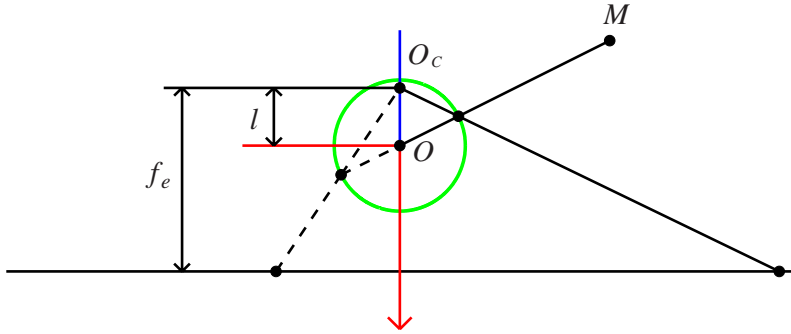


Fig. 5. A two-step projection via a unit sphere. O_C can lie inside, outside, or on the unit sphere.

2.3 Two-Step Projection via a Unit Sphere (TSP2)

The **TSP2** is the unified imaging model for central catadioptric and fisheye cameras. The reason for this will be given in Sect. 2.5. Obviously, the **TSP2** is a special case of the generalize model by setting $e \rightarrow 0, p \rightarrow \infty, ep \rightarrow 1$ (see Fig. 5). If we let $d_C = l$, the projection can be represented as:

$$\begin{aligned} (x, y) &= S_{l, f_e}(X, Y, Z) = Q_{0, \infty, f_e, l}(X, Y, Z) = \left(\hat{Q} \cup \bar{\hat{Q}} \right)_{0, \infty, f_e, l}(X, Y, Z) \\ &= \left(\frac{f_e X}{\pm l \sqrt{X^2 + Y^2 + Z^2} + Z}, \frac{f_e Y}{\pm l \sqrt{X^2 + Y^2 + Z^2} + Z} \right). \end{aligned} \quad (14)$$

The unit sphere used here is called the viewing sphere. O_C can lie inside, outside, or on the viewing sphere. Note that this model has been proposed for central catadioptric cameras by Geyer and Daniilidis [12] where O_C is only inside, or on the viewing sphere.

2.4 Two-Step Projection via a Quadric Surface from an Orthographic Camera (TSP3)

The **TSP3** is a special case of the generalized model with $f_e \rightarrow \infty, d_C \rightarrow \infty, \frac{f_e}{d_C} \rightarrow 1$ (see Fig. 6). The **TSP3** is also a special case of the model proposed in [20] where here we restrict that the quadric surface is a revolution of a conic section and the effective viewpoint is located at one of the foci of the conic section (Experimental results in Sect. 3 show that the special model with this restriction can also fit well for actual fisheye cameras). If we denote e as ε in order to avoid notational ambiguity where e has been used in the **TSP0** and **TSP1**, we can obtain:

$$\begin{aligned} (x, y) &= Q_{\varepsilon, p, \infty, \infty}(X, Y, Z) = \left(\hat{Q} \cup \check{Q} \cup \bar{\hat{Q}} \cup \bar{\check{Q}} \right)_{\varepsilon, p, \infty, \infty}(X, Y, Z) \\ &= \left(\pm \frac{pX}{\pm \frac{1}{\varepsilon} \sqrt{X^2 + Y^2 + Z^2} + Z}, \pm \frac{pY}{\pm \frac{1}{\varepsilon} \sqrt{X^2 + Y^2 + Z^2} + Z} \right). \end{aligned} \quad (15)$$

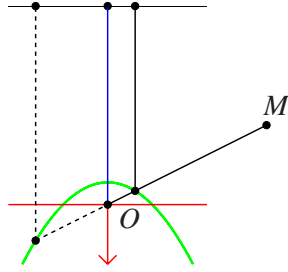


Fig. 6. A two-step projection via a quadric surface from an orthographic camera. The quadric surface can be ellipsoidal, paraboloidal, hyperboloidal, et al.

2.5 Equivalence among These Imaging Models

From (9) and (10) in Sect. 2.1, we know that the generalized model can be represented by a model with two parameters α, β :

$$\begin{aligned} (x, y) &= G_{\alpha, \beta}(X, Y, Z) \\ &= \left(\frac{\beta X}{\alpha \sqrt{X^2 + Y^2 + Z^2} + Z}, \frac{\beta Y}{\alpha \sqrt{X^2 + Y^2 + Z^2} + Z} \right), \end{aligned} \quad (16)$$

where $\alpha \in R, \beta \in R$ and $\beta \neq 0$. For the **TSP1**, from (11), (12) and (13) we obtain, $-1 \leq \alpha \leq 0 \cup \alpha = 1$ (for planar catadioptric camera, $e \rightarrow \infty, \alpha \rightarrow 0$). For the **TSP2**, from (14), we obtain $\alpha = \pm l$, obviously $\alpha \in R$ here. For the **TSP3**, from (15), we can obtain $\alpha = \frac{1}{\varepsilon}$. Let $\varepsilon \rightarrow \infty, \frac{1}{\varepsilon} \rightarrow 0$, and $\varepsilon \rightarrow 0, \frac{1}{\varepsilon} \rightarrow \infty$, then we obtain $\alpha \in R$.

Definition. $(x_1, y_1) = G_{\alpha_1, \beta_1}(X, Y, Z)$ and $(x_2, y_2) = G_{\alpha_2, \beta_2}(X, Y, Z)$ are two instances of the generalized two-step projection. If $\alpha_1 = \pm \alpha_2$ and $\beta_1 = s\beta_2 (s \in R, s \neq 0)$, we call that G_{α_1, β_1} and G_{α_2, β_2} are equivalent, and denoted by $G_{\alpha_1, \beta_1} \sim G_{\alpha_2, \beta_2}$.

From the above discussions, we have the following proposition for the equivalence among these models.

Proposition. $P_0 / \sim = P_2 / \sim = P_3 / \sim \supset P_1 / \sim$.

where P_i represents the set of all projections belong to **TSPi** ($i=0,1,2,3$), and P_i / \sim represents the quotient set¹ of P_i .

Let us assume $l \geq 0$. The equivalence among **TSP1**, **TSP2** and **TSP3** are shown in Table 1. Therefore, we can let the **TSP2** as the unified imaging model for central catadioptric and fisheye cameras (the fitness of the unified model for actual fisheye cameras is illustrated by experimental results in Sect. 3). The parameter l is called the characteristic parameter of the unified model. Obviously, central catadioptric and fisheye cameras can be classified by the characteristic parameter. Note that Geyer and Daniilidis [12] have proved the equivalence between the **TSP1** and the **TSP2** when $0 \leq l \leq 1$.

¹ The set consisting of all equivalence classes of \sim .

Table 1. Equivalence among **TSP1**, **TSP2** and **TSP3**, see text for details

TSP1	TSP2	TSP3
Ellipsoidal $0 < e < 1$	Inside the viewing sphere $0 < l < 1$	Hyperboloidal $\varepsilon > 1$
Paraboloidal $e = 1$	On the viewing sphere $l = 1$	Paraboloidal $\varepsilon = 1$
Hyperboloidal $e > 1$	Inside the viewing sphere $0 < l < 1$	Hyperboloidal $\varepsilon > 1$
	Outside the viewing sphere $l > 1$	Ellipsoidal $0 < \varepsilon < 1$

For the case of $l = 0$, we obtain:

$$(x_0, y_0) = S_{0, f_{e0}}(X, Y, Z) = \left(f_{e0} \frac{X}{Z}, f_{e0} \frac{Y}{Z} \right). \quad (17)$$

It is a perspective projection and corresponds to a pinhole camera or a central planar catadioptric camera. For the case of $l = 1$, we obtain:

$$(x_1, y_1) = S_{1, f_{e1}}(X, Y, Z) = \left(\frac{f_{e1}X}{\sqrt{X^2 + Y^2 + Z^2} + Z}, \frac{f_{e1}Y}{\sqrt{X^2 + Y^2 + Z^2} + Z} \right). \quad (18)$$

It is the stereographic projection and equivalent to a central para-catadioptric camera, or some fisheye cameras such as those used in [5] and [8]. The reason for the equivalence of a central para-catadioptric camera can be known from (12). The reason for the equivalence of some fisheye camera will be derived below. From Fig. 2, we have:

$$r' = \sqrt{x_0^2 + y_0^2}, r = \sqrt{x_1^2 + y_1^2}. \quad (19)$$

From (17),(18) and (19), we obtain:

$$r' = 2 \frac{f_{e0}}{f_{e1}} \frac{r}{1 - \frac{1}{f_{e1}^2} r^2}. \quad (20)$$

Obviously, (20) has the same form as (1).

3 Calibration

Under the unified imaging model, a line in space is projected to a conic in the image plane, and such a conic is called a line image. Since the unified imaging model is an extension of the unified imaging model for central catadioptric camera, the calibration methods for central catadioptric cameras using line images [2, 11,13,25] can be directly applied to fisheye cameras. These calibration methods for central catadioptric camera are all metric. Therefore, the metric calibration from single fisheye image only using projections of lines becomes possible via

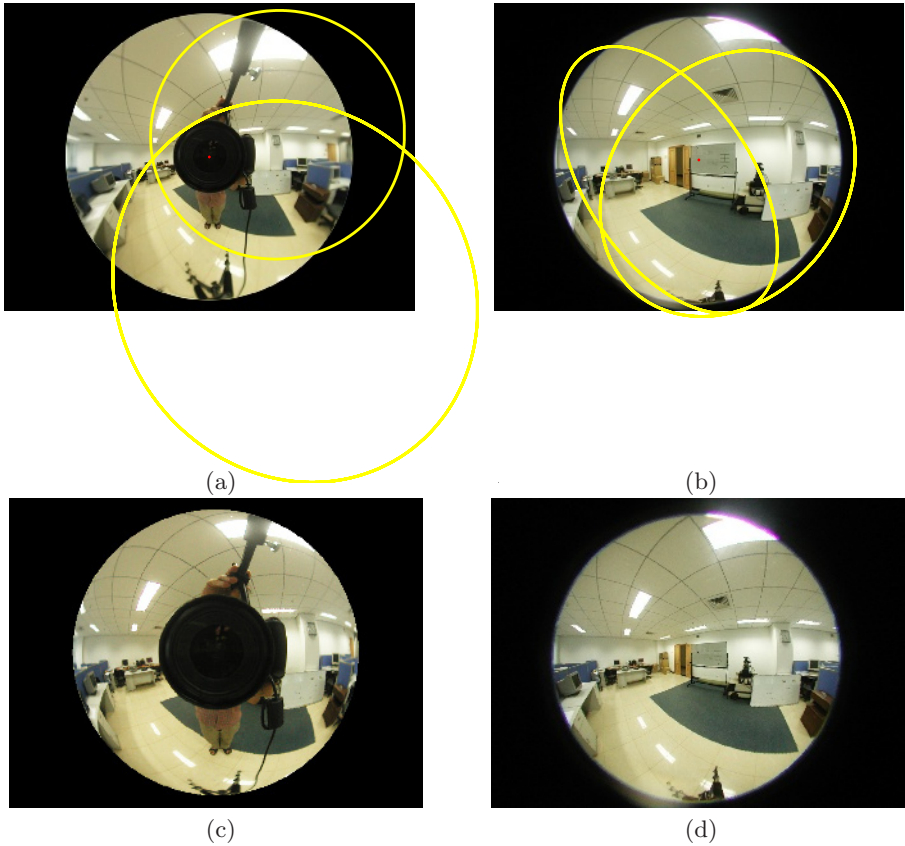


Fig. 7. Calibration of central catadioptric and fisheye cameras from projections of lines. (a) and (b) are conic fitting results for projections of lines from Fig. 1a and 1b. (c) and (d) are two synthesis images transformed from (a) and (b) respectively. The transformation from (a) to (c) is from catadioptric to fisheye, and the transformation from (b) to (d) is from fisheye to catadioptric. We can find that (a) and (d), (b) and (c) are almost the same (please compare the curvatures of corresponding projections of lines in these images). Note that the FOV of the catadioptric camera is larger than that of the fisheye camera.

our unified model but the existing methods for fisheye cameras in the literatures till now are all non-metric under the same conditions. Here, we use the calibration method based on the geometric invariants of line images proposed in [25], since the characteristic parameter l can be determined in an explicit form. Fig. 7 are some calibration results from the central catadioptric image and the fisheye image shown in Fig. 1a and 1b. Fig. 7a and 7b are conic fitting results for line images from Fig. 1a and 1b using some conic fitting methods [9,26]. Then we perform calibration using the method proposed in [25], and the intrinsic parameters of these cameras and the characteristic parameter l are obtained. The FOV of fisheye camera is also estimated at 189 ± 3.5 degrees which is very close to 183 degrees that provided by camera producer.

We know that the characteristic parameter l for central catadioptric cameras in the unified model is from 0 to 1, however we find the characteristic parameter l for some actual fisheye cameras is from 1 to infinite (for example, Nikon FC-E8 we used). Although the image formations with $0 < l < 1$ or $l > 1$ possess some similar properties, such as line images are conics, there are several different properties between them which are listed as follows: For central catadioptric camera with $0 < l < 1$, a line image can belong to any type of conic, namely, line, circle, ellipse, hyperbola and parabola, but for fisheye camera with $l > 1$, it can only be line, circle and ellipse. Another notable difference is that if a line image from central catadioptric camera is an ellipse, its major axis must pass through the image center. However if a line image from fisheye camera, its minor axis goes through the image center instead (see Fig. 7a and 7b).

Similar to central catadioptric image, we can determine the directions of captured lighting rays from fisheye image if the fisheye camera has been calibrated metrically. Examples of transformations between a fisheye image and a central catadioptric image based on the unified model are shown in Fig. 7cd. The transformation method used here is similar to the one proposed in [23]. Note that these transformations can be accomplished only after these images are metrically calibrated. We can find that Fig. 7a and 7d, Fig. 7b and 7c are almost same without noticeable difference except the FOV of the catadioptric camera is larger than that of the fisheye camera. Therefore, we can say that the results of metric calibration of the fisheye camera are comparable with those of the central catadioptric camera.

In order to validate the fitness of the unified model for many existing fisheye cameras, distortion correction procedures are performed for some fisheye images taken from previous publications [7,14,19,20,22,24]. We first calibrate fisheye cameras from these fisheye images, and then transform them into perspective ones. We find that distortion correction results using our unified model are comparable with those using existing models. All these demonstrate that the unified model fit well for these actual fisheye cameras.

4 Conclusions

We present a unified imaging model for central catadioptric and fisheye cameras. The unified imaging model is an extension of the unified imaging model for central catadioptric camera proposed by Geyer and Daniilidis. In order to prove the equivalence among imaging models, we present a generalized two-step projection via a quadric surface. Then other two-step projection models can be treated as the special cases of the generalized model. We show that the unified model can cover some existing models for fisheye camera, and fit well for many real fisheye cameras. The advantage of proposing the unified model is that many existing calibration methods for central catadioptric cameras can be directly applied to fisheye cameras. Furthermore, the metric calibration from single fisheye image only using projections of lines becomes possible with the unified model whereas the existing methods for fisheye cameras in the literatures till now are all non-metric under the same conditions.

References

1. S. Baker and S.K. Nayar, A Theory of Catadioptric Image Formation, In Proc. International Conference on Computer Vision, India, 1998, pp. 35–42
2. J.P. Barreto and H. Arajo, Geometric Properties of Central Catadioptric Line Images, In Proc. European Conference on Computer Vision, 2002, pp. 237–251
3. A. Basu and S. Licardie, Alternative models for fish-eye lenses, Pattern Recognition Letters, 16(4), 1995, pp. 433–441
4. M. Born and E. Wolf, Principles of Optics, Pergamon Press, 1965
5. C. Bräuer-Burchardt and K. Voss. A new algorithm to correct fish-eye- and strong wide-angle-lens-distortion from single images. In Proc. ICIP, pp. 225–228, 2001
6. D.C. Brown. Close range camera calibration. Photogrammetric Engineering, 37(8): pp.855–866, 1971
7. F. Devernay, O. Faugeras, Straight Lines Have to Be Straight: Automatic Calibration and Removal of Distortion from Scenes of Structured Environments, Machine Vision and Applications, 2001, vol.1, pp.14–24
8. A. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. Proceedings of IEEE Conference on CVPR, 2001
9. A. Fitzgibbon, M. Pilu, R. Fisher, Direct least-square fitting of ellipses, ICPR, 1996
10. M.M. Fleck, Perspective Projection: the Wrong Imaging Model, technical report 95-01, Computer Science, University of Iowa, 1995
11. C. Geyer and K. Daniilidis, Catadioptric Camera Calibration. ICCV 1999: 398–404
12. C. Geyer and K. Daniilidis, A Unifying Theory for Central Panoramic Systems and Practical Implications, In Proc. ECCV, 2000, pp. 445–462
13. C. Geyer and K. Daniilidis, Paracatadioptric Camera Calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(5): pp. 687–695
14. S. B. Kang, Radial distortion snakes, IAPR Workshop on MVA, 2000, pp. 603–606
15. B. Micusik and T. Pajdla, Estimation of Omnidirectional Camera Model from Epipolar Geometry, CVPR, 2003
16. K. Miyamoto. Fish eye lens. Journal of Optical Society of America, 54: pp. 1060–1061, 1964
17. S. K. Nayar, Omnidirectional Vision, Proc. of Eight International Symposium on Robotics Research, Shonan, Japan, October 1997
18. S.A. Nene and S.K. Nayar, Stereo with mirrors, In Proc. International Conference on Computer Vision, India, 1998, pp. 1087–1094
19. S. Shah, J. K. Aggarwal. Intrinsic Parameter Camera Calibration Procedure for a (High Distortion) Fish-Eye Lens Camera with Distortion Model and Accuracy Estimation. Pattern Recognition, 1996, vol.29, no.11, pp. 1775–1788
20. P.W. Smith, K.B. Johnson, and M.A. Abidi, Efficient Techniques for Wide-Angle Stereo Vision using Surface Projection Models, CVPR, 1999
21. T. Svoboda, T. Padjla, and V. Hlavac, Epipolar geometry for panoramic cameras, In Proc. European Conference on Computer Vision, 1998, pp. 218–231
22. R. Swaminathan, S.K. Nayar. Non-Metric Calibration of Wide-Angle Lenses and Polycameras. PAMI, 2000, pp. 1172–1178
23. M. Urban, T. Svoboda, T. Pajdla, Transformation of Panoramic Images: from hyperbolic mirror with central projection to parabolic mirror with orthogonal projection, Technical report, The Center for Machine Perception, Czech Technical University, Prague, 2000
24. Y. Xiong, K. Turkowski. Creating Image-Based VR Using a Self-Calibrating Fish-eye Lens. Proceedings of CVPR, 1997, 237–243

25. X. Ying, Z. Hu, Catadioptric Camera Calibration Using Geometric Invariants, International Conference on Computer Vision (ICCV2003), Nice, France, 2003
26. Z. Zhang, Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting, INRIA Raport de Recherche n 2676, October 1995

Image Clustering with Metric, Local Linear Structure, and Affine Symmetry

Jongwoo Lim¹, Jeffrey Ho², Ming-Hsuan Yang³,
Kuang-chih Lee¹, and David Kriegman²

¹ University of Illinois at Urbana-Champaign, Urbana, IL 61801

² University of California at San Diego, La Jolla, CA 92093

³ Honda Research Institute, Mountain View, CA 94041

Abstract. This paper addresses the problem of clustering images of objects seen from different viewpoints. That is, given an unlabelled set of images of n objects, we seek an unsupervised algorithm that can group the images into n disjoint subsets such that each subset only contains images of a single object. We formulate this clustering problem under a very broad geometric framework. The theme is the interplay between the geometry of appearance manifolds and the symmetry of the 2D affine group. Specifically, we identify three important notions for image clustering: the L^2 distance metric of the image space, the local linear structure of the appearance manifolds, and the action of the 2D affine group in the image space. Based on these notions, we propose a new image clustering algorithm. In a broad outline, the algorithm uses the metric to determine a neighborhood structure in the image space for each input image. Using local linear structure, comparisons (affinities) between images are computed only among the neighbors. These local comparisons are agglomerated into an affinity matrix, and a spectral clustering algorithm is used to yield the final clustering result. The technical part of the algorithm is to make all of these compatible with the action of the 2D affine group. Using human face images and images from the COIL database, we demonstrate experimentally that our algorithm is effective in clustering images (according to object identity) where there is a large range of pose variation.

1 Introduction

Given a collection of images, one may wish to group or cluster the images according to many different attributes of the images and their content. For instance, one may wish to cluster them based on some notion of human categories or taxonomies of objects. Or one might wish to cluster based on scene content (e.g., beach, agricultural, or urban scenes). Or perhaps one might wish to cluster all images into groups with the same lighting or with the same pose (this might only be relevant for images from a specific class such as faces [1]). In this paper, we consider the problem of clustering images according to the identity of the 3D objects, but where the observer's viewpoint has varied between images.

Clearly, this type of image clustering problem requires understanding how the images of an object vary under different viewing conditions, and so the goal of the clustering algorithm is to detect some consistent patterns among the images. A traditional computer vision approach to solve this problem would most likely include some kind of image feature extraction, e.g., texture, shape, filter bank outputs, etc. [2,3]. The underlying assumption is that some global or local image properties of a 3D object exist over a wide range of viewing conditions. The drawback of such an approach is that it is usually difficult to extract these features reliably and consistently. Appearance-based approaches e.g. [4,5] offer a different kind of strategy for tackling the clustering problem. For this type of algorithm, image feature extraction no longer plays a significant role. Instead, it is the geometric relations among images in the image space that is the focus of attention. The geometric concept that is central to appearance-based methods is the idea of an appearance manifold introduced in [6].

Our goal is to identify certain crucial geometric elements, such as the appearance manifold, that are central to the image clustering problem and to formulate a new clustering algorithm accordingly. Specifically, the two main contributions of this paper are:

1. We formulate the image clustering problem under a very general geometric framework. Using this framework, we provide a clear geometric interpretation of our algorithm and comparisons between our work and previous image clustering algorithms.
2. Motivated by geometric considerations, we propose a new image clustering algorithm.

We have tested our algorithm on two types of image data: images in the Columbia COIL database and images of human faces. Images of the 3D objects in the COIL database have more variation in surface texture and shape. Therefore, local image features can be extracted more reliably from these images [2]. For images of human faces, the variations in texture and shape are much more limited, and any clustering algorithm employing feature extractions is not expected to do well. We will show that our algorithm is capable of producing good clustering results for both types of image data.

2 Clustering Algorithm

In this section, we detail our image clustering algorithm. Schematically, our algorithm is similar to other clustering algorithms proposed previously, e.g., [7, 4]. That is, we define affinity measures between all pairs of images. These affinity measures are represented in a symmetric $n \times n$ matrix $A = (a_{ij})$, i.e., the affinity matrix and a straightforward application of any standard spectral clustering method [8,9] then yields our clustering result. The machinery employed to solve the clustering problem, i.e. spectral clustering, has been studied quite intensively in combinatorial graph theory [10], and it is of no concern to us here. Instead, our focus is on 1) explaining the geometric motivation behind our algorithm and 2) the definition of the affinity a_{ij} .

First, we define our image clustering problem. The input of the problem is a collection of unlabelled images $\{I_1, \dots, I_n\}$ and the number of clusters N . We assume that all images have the same number of pixels s , and by rasterizing the images, we obtain a collection of corresponding sample points $\{x_1, \dots, x_n\}$ in \mathbb{R}^s . Our algorithm outputs a cluster assignment for these images $\rho: \{I_1, \dots, I_n\} \rightarrow \{1, \dots, N\}$. Two images I_i and I_j belong to the same cluster if and only if $\rho(I_i) = \rho(I_j)$. A cluster, in our definition, consists of only images of one object. We further assume that the images of a cluster are acquired at different view points but under the same ambient illumination condition.

The problem so formulated is extremely general and without any further information, there is almost no visible structure to base the algorithm on. One obvious structure one can utilize is the ambient distance metric of the image space. The usual L^2 metric or its derivatives (affine-invariant L^2 distance or weighted L^2 distance) are such examples. By considering images as points in \mathbb{R}^s , we are naturally led to the notion of appearance manifolds [6]. Accordingly, the input images imply the existence of N sub-manifolds of \mathbb{R}^s , $\{M_1, \dots, M_N\}$ such that two points x_i, x_j belong to the same cluster if and only if $x_i, x_j \in M_k$ for some $1 \leq k \leq N$, with each M_i denoting the appearance manifold of an object. Implicit in the concept of appearance manifolds is the idea of local linearity. That is, if x_1, \dots, x_l are points belonging to the same cluster and if they are sufficiently close according to the distance metric, then each point x_i can be well-approximated linearly by its neighbors: $x_k \approx \sum_{j \neq k} a_j x_j$ for some real numbers a_j .

Metric and local linearity are two very general geometric notions and they do not pertain only to image clustering problems. It is the action of the 2D affine group G ¹ that characterizes our problem as an image clustering problem rather than a general data clustering problem. If $\{x_1, \dots, x_n\}$ were data of a different sort, e.g. data from a meteorological or high energy physics experiment, there will not be an explicit action of G . It is precisely because the 2D nature of the images and the way we rasterize the image to form points in \mathbb{R}^s , we can explicitly calculate the action of G given a sample point x . In particular, each appearance manifold M_i is invariant under G , i.e. if $x \in M_i$ then $\gamma(x) \in M_i$ for each $\gamma \in G$. In this sense, the clustering problem acquires a symmetry played by the 2D affine group². In summary, we have identified three important elements to the image clustering problem. First, there is the ambient L^2 (and its derivatives) metric of the image space. Second, each cluster has local linear structure. The metric and local linearity are the only two geometric structures we can utilize in designing the algorithm. The third element is the affine symmetry of the problem. Our challenge is to design a clustering algorithm that takes into account these three elements. In a very general outline, what is needed is to design metric and local linear structure that are both invariant under the affine group G and to seek an interesting and effective coupling between the metric and

¹ Henceforth, except at a few places, G will invariably denote the 2D affine group.

² Strictly speaking, the symmetry group will depend on what type of imaging model is used for the problem. In general, it will be a subgroup of G rather than G itself.

linear structure, which are two rather disparate geometric notions. Surprisingly, using only these three very general structures, we can formulate a clustering algorithm which will be demonstrated to be effective for a variety of image clustering problems. Our algorithm is compact and purely computational. Many standard vision techniques, such as local feature extractions and PCA, will not make their appearances in our algorithm.

The clustering problem we studied here is considerably more difficult than the illumination clustering studied in [11]. The main difference between their case and ours is a difference between global and local. In the illumination case, the linear structure, the illumination cone, is a global structure and it can be exploited directly in designing the clustering algorithm. In our case, the linear structure is only a local structure and unlike the cone which admits a compact and precise description via its generators, our local linear structure is more difficult to quantify. Therefore, the exploitation of the local linear structure in our algorithm is more subtle than in the illumination case.

2.1 Metric Structure

Since the input images are considered as a collection of points in \mathbb{R}^s , the usual L^2 -distance metric and its derivatives offer the simplest affinity measures between a pair of data points. However, since the clusters form manifolds in \mathbb{R}^s , they are not expected to localize in some region of \mathbb{R}^s independently of other clusters. This observation can be supported by the fact that the Euclidean distance between two face images of different identities acquired at the same pose is almost always smaller than the Euclidean distance of two images of the same identity but acquired at different poses [12,13]. Two analogous situations in 3D are depicted in Figure 1. They clearly demonstrate that if the metric information is used for defining affinity, then "medium" and "long-distance" comparisons are usually erroneous.

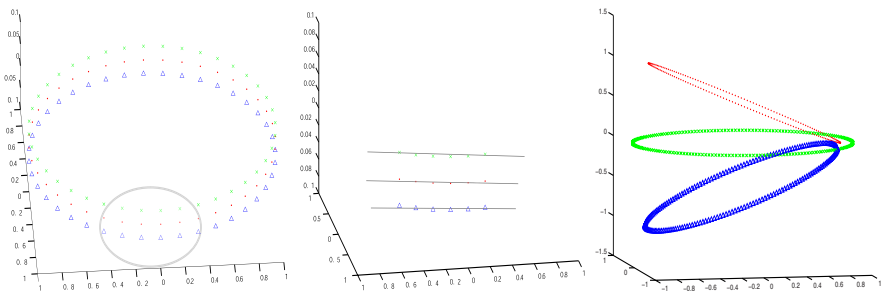


Fig. 1. Left(A) Three parallel circles. The points on each circle are uniformly sampled and the distance between adjacent circles is slightly smaller than the distance between two neighboring points on the same circle. **Center(B)** A "magnified" view of a neighborhood. **Right(C)** The top and bottom circles are rotated by $\pm 30^\circ$.

However, Figure 1(B) suggests one good way of using the metric is not to use it directly for comparison. Instead, we can use the metric to pick data points for which the comparisons will be made. In particular, for each point x , the metric defines a neighborhood and in this neighborhood, non-metrical information can be exploited to do the comparison (i.e., defining affinity). In this way, the metric defines a collection of local clustering problems, and the affinities computed in these local settings will then be put into the global affinity matrix to provide a final clustering result.

2.2 Local Linear Structure (LLS)

Figure 1 shows two examples which are unlikely to be clustered correctly using the metric information along. Figure 1(A) is a good example. The data collection contains points sampled uniformly from three circles in \mathbb{R}^3 . The distance between adjacent circles are slightly smaller than the distance between two neighboring points on the same circle. To the best of our effort, we can not correctly cluster the data into three circles using only metric information. The point of course is that the manifold structure of the circles must be taken into consideration. One possible way to use the manifold structure is to compute the "tangent space" at each sample point using Principal Component Analysis in a neighborhood of the sample point, as in [4]. This approach can correctly cluster Figure 1(A) but unlikely³ to cluster Figure 1(C) correctly. This is mainly because the local linear estimate using PCA becomes unstable in the region when the circles come into close contact with each other.

Instead of working with tangents, we shift our focus slightly to consider the secant approximation of a sample point by its neighbors, see Figure 2(A). For a smooth 2D curve, each point x can be approximated well by a point on the secant chord formed by two of its sufficiently close neighbors y_1, y_2 : $x \approx a_1 y_1 + a_2 y_2$ with a_1, a_2 non-negative and $a_1 + a_2 = 1$. This can be generalized immediately to higher dimension: for a point x and its neighbors, $\{y_1, \dots, y_K\}$, we can try to compute a set of non-negative coefficients ω_i which is the solution to the following optimization problem:

$$\min \left\| x - \sum_{i=1}^K \omega_i y_i \right\|_{L^2}^2 \quad (1)$$

with the constraint that $\sum_{i=1}^K \omega_i = 1$. Assuming $\{y_1, \dots, y_K\}$ are linearly independent⁴, then the coefficients ω_i are unique. Figure 2(B) illustrates that the magnitude of the coefficients ω_i can be used as an affinity measure locally to detect the presence of any linear structure. That is, a large magnitude of ω_i indicates the possibility that y_i and x share a common local linear structure.

³ To the best of our effort!

⁴ In the image space \mathbb{R}^s , this is almost always true since $K \ll s$.

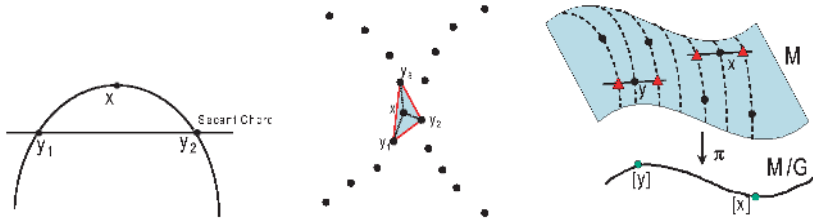


Fig. 2. Left(A) The secant chord approximation of a point on a smooth curve by its neighbors. **Center(B)** Two semi-circles. There are three possible secant chord approximations of x by the three sides of the shaded triangle. **Right(C)** The shaded surface denotes the appearance manifold M and the dashed lines are the orbits of the affine group G . The projection map π sends each point x in M to the corresponding point $[x] \in M/G$. The solid circles denote the sample points. In order to construct the local linear structure in M/G , we have to move the sample points along the orbits to produce "virtual" samples, denoted by the triangles.

Applying the idea we have outlined so far, a simple data clustering algorithm can be designed⁵, and we can cluster all the examples above correctly. On the other hand, to our best effort, we can't find a simple and straightforward algorithm, based on the more traditional clustering techniques such as the K-means and connected component analysis, etc., that can successfully cluster all of these examples.

2.3 Affine Symmetry and Quotient Spaces

As we mentioned earlier, the presence of the 2D affine group distinguishes the image clustering problem from the general data clustering problem. The task now is to put both the metric and local linear structure into an affine invariant setting (as best as we can). Affine invariant L^2 metric and many of its variants have been studied before in the literature [5,14,15], etc. Our effort is to propose a method for defining local linear structures that are affine invariant; in particular, we want to reformulate Equation 1 in an affine-invariant way.

We will explain this with the mathematical notion of a quotient space [16]. In general, when there is a group G acting on a manifold M , one can associate this action with an abstract topological space M/G , the quotient space. Loosely speaking, the space M/G parameterizes the orbits of the group action. See Figure 2(C). The important point is that any quantity defined in M that is invariant under the G -action can be naturally defined as a derived quantity in the space M/G . For instance, if we have a G -invariant metric on M , this metric then in turn defines a metric on M/G .

Specializing to our clustering problem, the manifold M is the union of the appearance manifolds, $\{M_1, \dots, M_N\}$, and the group G is the 2D affine group.

⁵ Compute ω_i for each sample point using its K -nearest neighbors provided by the metric. Form a symmetric affinity matrix using ω_i and apply the spectral clustering.

We have the natural projection map $\pi : M \rightarrow M/G$ which takes each point x of M to the point $[x] \in M/G$ which parameterizes the orbit containing x . The manifolds $\{M_1, \dots, M_N\}$ now descend down to M/G to form $\{\tilde{M}_1, \dots, \tilde{M}_N\}$. By speaking of affine invariant local linear structure, we are speaking of the local linear structures of these "manifolds" ⁶, $\{\tilde{M}_1, \dots, \tilde{M}_N\}$.

To compute the local linear structures of $\{\tilde{M}_1, \dots, \tilde{M}_N\}$, we can mimic the standard slice construction for quotient spaces [16]. The idea is that for each point $[x]$ of M/G , we can compute its local linear structure by lifting the computation to a sample point $x \in M$ such that $\pi(x) = [x]$. At each such point x , we take a "slice" of the group action, i.e., a linear subspace centered at x that is orthogonal to the G -action through x and we analyze the local linear structures on the slice. See Figure 2.(C). For each sample point x we find a slice S . We project all other sample points down to S using G , i.e. for a sample point y , we find a $\gamma \in G$ such that $\gamma(y) \in S$. Note that such γ may not exist for every y but we only need a few such y s to characterize a neighborhood of x . Let $\{y'_1, \dots, y'_s\}$ be the projected points on S . We use the L^2 metric in S to select the right neighbors of x , say, $\{y'_1, \dots, y'_K\}$ and use them in defining the local linear structure at $[x]$ via Equation 1.

In the actual implementation, we modify the slice construction outlined above. Instead of actually computing the subspace S , we determine the K neighbors $\{y'_1, \dots, y'_K\}$ by using the "one-sided distance" [14]. For each input sample y , the "one-sided distance" is defined as

$$d_G(x, y) = \min_{\gamma \in G} \left\{ \min \left\{ \|x - \gamma(y)\|_{L^2}^2, \|y - \gamma(x)\|_{L^2}^2 \right\} \right\}.$$

Although $d_G(x, y)$ is not a metric, it still allows us to define the K -nearest neighbors of x . The K neighbors $\{y'_1, \dots, y'_K\}$ of x above are just $\{\gamma_1(y_1), \dots, \gamma_K(y_K)\}$ with each γ_i minimizes the one-sided distance between x and y_i .

3 Related Work

In this section, we compare our algorithm with some of the well-known image clustering algorithms in the literature. Needless to say, the 2D affine group has a long history in the computer vision literature. In particular, intensive effort has been focused on studying (quasi-) affine invariant metric such as the tangent distance e.g. [15,17]. For image clustering, affine invariant metric has made its appearance in the work of Fitzgibbon and Zisserman [5,14]. Most of the effort in these two papers has been focused on designing an affine invariant metric that will be effective for clustering. In the language of the quotient space, they are

⁶ Demonstrating the quotient space is actually some "nice" geometric object is generally a very delicate mathematical problem [16]. It is not our intention here to rigorously define the space M/G . Our goal is to use the idea of quotient space to explain the motivation of the algorithm and in the next section, to compare our algorithm with other previous algorithms.

doing clustering on M/G using metric information alone. Our algorithm also uses the metric information in M/G but it also explicitly tries to cluster "manifolds" in M/G . Although good clustering results can be obtained by considering metric alone, we believe that by incorporating both the metric and local linearity, it offers 1) a more effective clustering algorithm and 2) a more complete geometric description of the clustering algorithm.

Another well-known image clustering algorithm that explicitly uses the concept of the appearance manifold is [4]. However, there are two major differences between our work and theirs. First, the affine symmetry is absent in [4]. One of the main themes of this paper is that the action of the 2D affine group is of central importance in formulating any image clustering problem. Second, there is an important difference between our concept of local linearity and theirs. In [4], the concept of local linearity is embodied in the idea of tangent space of the appearance manifold; therefore, PCA is used to estimate local linear subspaces. In contrast, our concept of local linearity is on how best the "neighbors" can linearly approximate a given sample point, and it is formulated through Equation 1. This concept of local linearity also allows non-geometric interpretation in terms of image comparisons using parts of objects as in [18]; however, it is not clear if there is a non-geometric interpretation of the tangent spaces used in [4].

[2,3] are two other interesting and related papers on image clustering. Their approaches and ours are fundamentally different in that our algorithm is completely image-based while their algorithms focus on extracting salient image features and incorporating more sophisticated machine learning techniques for clustering. However, comparisons between their experimental results and ours will be made in the next section.

4 Experiments

In this section, we report our experimental results. Our image clustering algorithm, as detailed in Figure 3, has been implemented in MATLAB. Two different types of image data were used to test the algorithm, images of 3D objects and images of human faces. Substantial variations in appearances are observed in all image datasets. The main difference between these two types of datasets is the variation in surface texture. For the former type, the surface texture varies greatly and local image features (such as corners) can be more reliably extracted. Human faces, on the other hand, have much limited variation in surface texture and local image features become less useful. Traditionally, these two different types of image data were attacked separately using feature-based methods (e.g. [2]) and appearance-based methods (e.g. [1]), respectively. However, the results below show that our algorithm is capable of obtaining good clustering results for both types of images.

Except for the affine-invariant metric $d_G(x, y)$, the implementation is straightforward and it follows closely the steps outlined in Figure 3. Given two images, I_1, I_2 , $d_G(I_1, I_2)$ is computed as follows. First, we define a Gaussian distribution p on 2D affine group centered at the identity. Since we only consider

1. Inputs

A collection of unlabelled images $\{I_1, \dots, I_n\}$. Considered the images as data points $\{x_1, \dots, x_n\}$ in the image space \mathbb{R}^s , and the number of clusters N .

2. Use Metric to Choose Neighbors

For each data point x , compute a set of K nearest neighbors using the distance measure $d_G(x, y)$ defined above.

3. Use local linear structure

For each x and its K -neighbors $\{x_1, \dots, x_K\}$ determined in the previous step, let $\{y_1, \dots, y_K\}$ be the points in \mathbb{R}^s such that $y_i = \gamma(x_i)$ for some $\gamma \in G$ and y_i minimizes the distances between x and all points on the orbit of G through x_i . Using y_i 's to linearly approximate x by determining a collection of K non-negative real numbers ω_i that minimizes the objective function

$$\left\| x - \sum_{i=1}^K \omega_i y_i \right\|_{L^2}^2, \quad \text{with the constraint that } \sum_{i=1}^K \omega_i = 1$$

4. Use ω_i as the affinity measure

Define an affinity measure d_Ω between two data points x_i and x_j : $d_\Omega(x_i, x_j) = \min(1/\omega_{ij}, 1/\omega_{ji})$ where ω_{ij} is the coefficient computed in the previous step for x_i . If x_j is not among the K -neighbors of x_i , ω_{ij} is set to 0. Apply the spectral clustering algorithm (e.g., [8]) using this affinity to yield the final clustering result.

Fig. 3. The clustering algorithm.

small affine corrections, p can be expressed in a local coordinates system centered at the identity by expressing each (small) affine transformation in terms of the usual six parameters (a 2x2 matrix plus translation). Using these six parameters, p is a Gaussian distribution with diagonal covariance matrix. Next, we determine an affine transformation γ such that it minimizes the function

$$E(\gamma) = \min \{ d_{L^2}(\gamma(I_1), I_2), d_{L^2}(I_1, \gamma(I_2)) \}$$

where d_{L^2} is the usual L^2 distance metric between two images. γ can be found using gradient descent [19]. $d_G(I_1, I_2)$ is then defined as the sum $E(\gamma) - \log p(\gamma)$. The reason for incorporating the Gaussian $p(\gamma)$ is to penalize “over-corrections” by large affine transformations [14].

4.1 Datasets

In this subsection, we fix the notations for various image datasets we used in the experiments and give brief descriptions of the datasets. For images of 3D objects, we use the COIL datasets from Columbia, which are popular datasets for validating object recognition algorithms. There are two COIL datasets, COIL20 and COIL100. They contain 20 and 100 objects, respectively. For both datasets, the images of each object were taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images. Since this sampling is quite dense, we “sub-sampled” the image collections to make clustering problem more interesting. We let COIL20.2 denote the collection of images obtained from COIL20

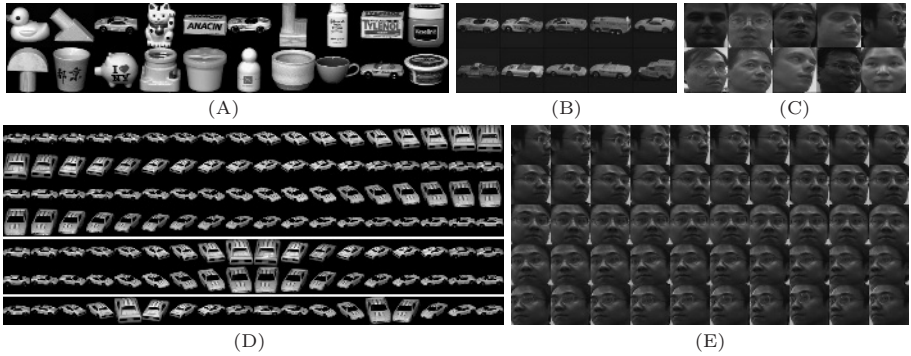


Fig. 4. (A): Representative images of objects in COIL20 (B): The ten vehicles in VEH10.2 (C): The ten individuals in FACE10 (D): Sampling frequency. First 4 rows are images of one object from COIL20, next 2 rows from COIL20.2, and last row from COIL20.4 (E): Pose variation in FACE10.

by sub-sampling it with a factor of 2. So COIL20.2 contains the same number of objects as the original COIL20 but with half as many images per object. Similarly, COIL20.4 denotes the collection obtained from COIL20 by sub-sampling it with a factor of 4 and so on. From COIL100.2, we placed all vehicle images in this collection together to form a new dataset, VEH10.2. The images of these vehicles have similar appearances and therefore, they offer a challenging dataset to test our algorithm. For images of human faces, we collected video sequences of ten individuals to form ten image sequences with each sequence containing 50 images. Pose variation in this collection is quite large and because of the differences in individual motion, the image sequences do not have uniform variation in pose. This dataset will be denoted by FACE10.

4.2 Results

The experimental results are reported in Table 1. As is clear from Table 1, our algorithm produces good clustering results for all datasets except COIL100.4. The algorithm’s performance on COIL100 is not surprising considering that there are 100 objects in COIL100.4 and the images are rather sparsely sampled (every 20 degrees). Error rates are calculated as the ratio of the number of misclustered images over the number of images⁷. The error rates are shown together with the parameter K which defines the size of the local neighborhoods. We also mention that there are clustering results on COIL20 database reported in [2]. We can not translate their definition of errors into ours. However, they do report non-zero error rate while our clustering algorithm achieves a perfect clustering result for the COIL20 dataset.

⁷ For each cluster emerged from the clustering result, we try to match it with the known clusters (ground-truth). Once the one-to-one map between the new clusters and known clusters is computed, the error ratio can be calculated accordingly. For instance, a random assignment of a collection of N clusters of equal size will produce an error rate of $\frac{N-1}{N}$ according to our definition.

Table 1. Clustering results of our algorithm

	FACE10	COIL20	COIL20.2	COIL20.4	VEH10.2	COIL100.2	COIL100.4
Error	0.00%	0.00%	5.14%	19.44%	11.11%	20.69%	34.89%
K	10	8	8	8	3	6	10

Table 2. Comparison with other clustering algorithms

Algorithms	Datasets			
	COIL20.2	COIL20.4	FACE10	VEH10.2
Our algorithm	5.14%	19.44%	0.00%	11.11%
Affine+K-NN+Spectral	7.36%	21.11%	13.00%	27.50%
Affine+Spectral	10.14%	25.83%	22.00%*	40.00%
Euclidean+Spectral	35.14%	33.06%	25.60%*	61.67%
Euclidean+K-means	39.58%	48.06%	46.00%	74.44%

* Spectral clustering results indicate the results may not be robust

4.3 Comparison with Other Clustering Algorithms

Table 2 lists the result of comparing (on four different datasets) our algorithm with some standard off-the-shelf algorithms. First, two standard clustering algorithms, K-means and spectral clustering algorithm [8] with the usual L^2 -distance metric, are compared with our results. It clearly demonstrates that direct L^2 comparisons without affine-invariance are not sufficient at all. Next, we incorporate affine-invariance but without using local comparisons (Affine+Spectral). This is the “one-sided” distance measure [14] and again, it is still not able to produce good clustering results. Next, we show that by incorporating local linear structure in the algorithm, it does indeed enhance the performance of the clustering algorithm. Note that in our framework, once a neighborhood structure has been determined, we exploit the local linear structure to cluster points in the neighborhood. To show that this is indeed effective and necessary, we replace this step of our algorithm with direct metric comparisons. That is, we are computing local affinities based purely on the “one-sided” distance measure (Affine+K-NN+Spectral). We expect that our algorithm will be an improvement over this method because of our use of non-metrical information, and the results do indeed corroborate our claim.

Finally, in Figure 5, we illustrate several results of our local linear estimates, i.e., the ω_i . Although the K -nearest neighbors of an image generally contain

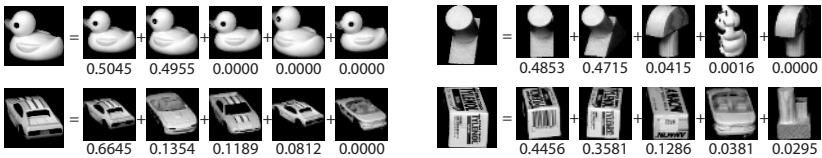


Fig. 5. Images, their neighbors and the local linear structure, ω_i 's.

images of other objects, in each case, ω_i correctly pick out the right images to form strong affinities.

5 Concluding Remarks

In this paper, we have proposed an image clustering algorithm, and we have demonstrated with a number of experiments that our algorithm is indeed effective for clustering images of 3D objects undergoing large pose variation. One obvious limitation of our algorithm is that we do not explicitly model the illumination effect. However, [11] has demonstrated that it is possible to cluster images with illumination variation using global linear structures. How best to incorporate our local structure and the global one in [11] into an effective image clustering algorithm that can deal with both lighting and pose variations will be a challenging and interesting research direction for the future.

Acknowledgements. This work was funded under NSF CCR 00-86094, NSF CCR 00-86094, the U.C. MICRO program, and the Honda Research Institute.

References

1. Li, S., Lv, X., Zhang, H.: View-based clustering of object appearances based on independent subspace analysis. In: Proceedings of IEEE International Conference on Computer Vision. (2001) 295–300
2. Saux, B.L., Boujemaa, N.: Unsupervised robust clustering for image database categorization. In: International Conference on Pattern Recognition. Volume 1. (2002) 259–262
3. Frigui, H., Boujemaa, N., Lim, S.: Unsupervised clustering and feature discrimination with application to image database categorization. In: Joint 9th IFSA World Congress and 20th NAFIPS Conference. (2001)
4. Basri, R., Roth, D., Jacobs, D.: Clustering appearances of 3D objects. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (1998) 414–420
5. Fitzgibbon, A.W., Zisserman, A.: On affine invariant clustering and automatic cast listing in movies. In Heyden, A., Sparr, G., Nielsen, M., Johansen, P., eds.: Proceedings of the Seventh European Conference on Computer Vision. LNCS 2353, Springer-Verlag (2002) 304–320
6. Murase, H., Nayar, S.K.: Visual learning and recognition of 3-D objects from appearance. In: International Journal of Computer Vision. Volume 14. (1995) 5–24
7. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 888–905
8. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In Ditterich, T., Becker, S., Ghahramani, Z., eds.: Advances in Neural Information Processing Systems 15, MIT Press (2002) 849–856
9. Weiss, Y.: Segmentation using eigenvectors: A unifying view. In: Proceedings of IEEE International Conference on Computer Vision. Volume 2. (1999) 975–982

10. Chung, F.R.K.: Spectral Graph Theory. American Mathematical Society (1997)
11. Ho, J., Yang, M.H., Lim, J., Lee, K.C., Kriegman, D.: Clustering appearances of objects under varying illumination conditions. In: IEEE Conf. on Computer Vision and Pattern Recognition. Volume 1. (2003) 11–18
12. Graham, D.B., Allinson, N.M.: Norm²-based face recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (1999) 586–591
13. Raytchev, B., Murase, H.: Unsupervised face recognition from image sequences based on clustering with attraction and repulsion. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Volume 2. (2001) 25–30
14. Fitzgibbon, A.W., Zisserman, A.: Joint manifold distance: a new approach to appearance based clustering. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2003) 26–33
15. Simard, P., Cun, Y.L., Denker, J., Victorri, B.: Transformation invariance in pattern recognition - tangent distance and tangent propagation. In: Neural Networks. Volume 1524. (1998) 239–274
16. Mumford, D., Kirwan, F., Fogarty, J.: Geometric Invariant Theory. Springer-Verlag (1994)
17. Frey, B., Jojic, N.: Fast, large-scale transformation-invariant clustering. In: Advances in Neural Information Processing Systems 14. (2001) 721–727
18. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. *Nature* **401** (1999) 781–791
19. Hager, G.D., Belhumeur, P.N.: Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20** (1998) 1025–1039

Face Recognition with Local Binary Patterns

Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen

Machine Vision Group, Infotech Oulu

PO Box 4500, FIN-90014 University of Oulu, Finland,

{tahonen,hadid,mkp}@ee.oulu.fi, <http://www.ee.oulu.fi/mvg/>

Abstract. In this work, we present a novel approach to face recognition which considers both shape and texture information to represent face images. The face area is first divided into small regions from which Local Binary Pattern (LBP) histograms are extracted and concatenated into a single, spatially enhanced feature histogram efficiently representing the face image. The recognition is performed using a nearest neighbour classifier in the computed feature space with Chi square as a dissimilarity measure. Extensive experiments clearly show the superiority of the proposed scheme over all considered methods (PCA, Bayesian Intra/extrapersonal Classifier and Elastic Bunch Graph Matching) on FERET tests which include testing the robustness of the method against different facial expressions, lighting and aging of the subjects. In addition to its efficiency, the simplicity of the proposed method allows for very fast feature extraction.

1 Introduction

The availability of numerous commercial face recognition systems [1] attests to the significant progress achieved in the research field [2]. Despite these achievements, face recognition continues to be an active topic in computer vision research. This is due to the fact that current systems perform well under relatively controlled environments but tend to suffer when variations in different factors (such as pose, illumination etc.) are present. Therefore, the goal of the ongoing research is to increase the robustness of the systems against different factors. Ideally, we aim to develop a face recognition system which mimics the remarkable capabilities of human visual perception. Before attempting to reach such a goal, one needs to continuously learn the strengths and weaknesses of the proposed techniques in order to determine new directions for future improvements. To facilitate this task, the FERET database and evaluation methodology have been created [3]. The main goal of FERET is to compare different face recognition algorithms on a common and large database and evaluate their performance against different factors such as facial expression, illumination changes, aging (time between the acquisition date of the training image and the image presented to the algorithm) etc.

Among the major approaches developed for face recognition are Principal Component Analysis (PCA) [4], Linear Discriminant Analysis (LDA) [5] and

Elastic Bunch Graph Matching (EBGM) [6]. PCA is commonly referred to as the "eigenface" method. It computes a reduced set of orthogonal basis vectors or eigenfaces of the training face images. A new face image can be approximated by a weighted sum of these eigenfaces. PCA provides an optimal linear transformation from the original image space to an orthogonal eigenspace with reduced dimensionality in the sense of least mean squared reconstruction error. LDA seeks to find a linear transformation by maximising the between-class variance and minimising the within-class variance. In the EBGM algorithm, faces are represented as graphs, with nodes positioned at fiducial points and edges labelled with distance vectors. Each node contains a set of Gabor wavelet coefficients, known as a jet. Thus, the geometry of the face is encoded by the edges while the grey value distribution (texture) is encoded by the jets. The identification of a new face consists of determining among the constructed graphs, the one which maximises the graph similarity function. Another proposed approach to face recognition is the Bayesian Intra/extrapersonal Classifier (BIC) [7] which uses the Bayesian decision theory to divide the difference vectors between pairs of face images into two classes: one representing intrapersonal differences (i.e. differences in a pair of images representing the same person) and extrapersonal differences.

In this work, we introduce a new approach for face recognition which considers both shape and texture information to represent the face images. As opposed to the EBGM approach, a straightforward extraction of the face feature vector (histogram) is adopted in our algorithm. The face image is first divided into small regions from which the Local Binary Pattern (LBP) features [8,9] are extracted and concatenated into a single feature histogram efficiently representing the face image. The textures of the facial regions are locally encoded by the LBP patterns while the whole shape of the face is recovered by the construction of the face feature histogram. The idea behind using the LBP features is that the face images can be seen as composition of micro-patterns which are invariant with respect to monotonic grey scale transformations. Combining these micro-patterns, a global description of the face image is obtained.

2 Face Description with Local Binary Patterns

The original LBP operator, introduced by Ojala *et al.* [9], is a powerful means of texture description. The operator labels the pixels of an image by thresholding the 3x3-neighbourhood of each pixel with the center value and considering the result as a binary number. Then the histogram of the labels can be used as a texture descriptor. See Figure 1 for an illustration of the basic LBP operator.

Later the operator was extended to use neighbourhoods of different sizes [8]. Using circular neighbourhoods and bilinearly interpolating the pixel values allow any radius and number of pixels in the neighbourhood. For neighbourhoods we will use the notation (P, R) which means P sampling points on a circle of radius of R . See Figure 2 for an example of the circular (8,2) neighbourhood.

Another extension to the original operator uses so called *uniform patterns* [8]. A Local Binary Pattern is called uniform if it contains at most two bitwise

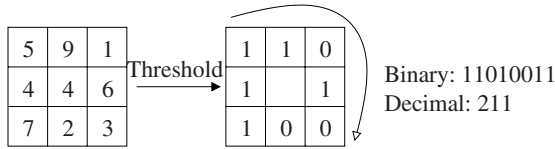


Fig. 1. The basic LBP operator.

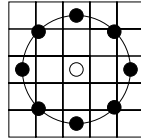


Fig. 2. The circular (8,2) neighbourhood. The pixel values are bilinearly interpolated whenever the sampling point is not in the center of a pixel.

transitions from 0 to 1 or vice versa when the binary string is considered circular. For example, 00000000, 00011110 and 10000011 are uniform patterns. Ojala *et al.* noticed that in their experiments with texture images, uniform patterns account for a bit less than 90 % of all patterns when using the (8,1) neighbourhood and for around 70 % in the (16,2) neighbourhood.

We use the following notation for the LBP operator: $LBP_{P,R}^{u2}$. The subscript represents using the operator in a (P, R) neighbourhood. Superscript $u2$ stands for using only uniform patterns and labelling all remaining patterns with a single label.

A histogram of the labeled image $f_l(x, y)$ can be defined as

$$H_i = \sum_{x,y} I \{f_l(x, y) = i\}, i = 0, \dots, n - 1, \quad (1)$$

in which n is the number of different labels produced by the LBP operator and

$$I \{A\} = \begin{cases} 1, & A \text{ is true} \\ 0, & A \text{ is false.} \end{cases}$$

This histogram contains information about the distribution of the local micropatterns, such as edges, spots and flat areas, over the whole image. For efficient face representation, one should retain also spatial information. For this purpose, the image is divided into regions R_0, R_1, \dots, R_{m-1} (see Figure 5 (a)) and the spatially enhanced histogram is defined as

$$H_{i,j} = \sum_{x,y} I \{f_l(x, y) = i\} I \{(x, y) \in R_j\}, i = 0, \dots, n - 1, j = 0, \dots, m - 1. \quad (2)$$

In this histogram, we effectively have a description of the face on three different levels of locality: the labels for the histogram contain information about the patterns on a pixel-level, the labels are summed over a small region to produce information on a regional level and the regional histograms are concatenated to build a global description of the face.

From the pattern classification point of view, a usual problem in face recognition is having a plethora of classes and only a few, possibly only one, training sample(s) per class. For this reason, more sophisticated classifiers are not needed but a nearest-neighbour classifier is used. Several possible dissimilarity measures have been proposed for histograms:

- Histogram intersection:

$$D(\mathbf{S}, \mathbf{M}) = \sum_i \min(S_i, M_i) \quad (3)$$

- Log-likelihood statistic:

$$L(\mathbf{S}, \mathbf{M}) = - \sum_i S_i \log M_i \quad (4)$$

- Chi square statistic (χ^2):

$$\chi^2(\mathbf{S}, \mathbf{M}) = \sum_i \frac{(S_i - M_i)^2}{S_i + M_i} \quad (5)$$

All of these measures can be extended to the spatially enhanced histogram by simply summing over i and j .

When the image has been divided into regions, it can be expected that some of the regions contain more useful information than others in terms of distinguishing between people. For example, eyes seem to be an important cue in human face recognition [2,10]. To take advantage of this, a weight can be set for each region based on the importance of the information it contains. For example, the weighted χ^2 statistic becomes

$$\chi_w^2(\mathbf{S}, \mathbf{M}) = \sum_{i,j} w_j \frac{(S_{i,j} - M_{i,j})^2}{S_{i,j} + M_{i,j}}, \quad (6)$$

in which w_j is the weight for region j .

3 Experimental Design

The CSU Face Identification Evaluation System [11] was utilised to test the performance of the proposed algorithm. The system follows the procedure of the FERET test for semi-automatic face recognition algorithms [12] with slight modifications. The system uses the full-frontal face images from the FERET database and works as follows (see Figure 3):

1. The system preprocesses the images. The images are registered using eye coordinates and cropped with an elliptical mask to exclude non-face area from the image. After this, the grey histogram over the non-masked area is equalised.
2. If needed, the algorithm is trained using a subset of the images.

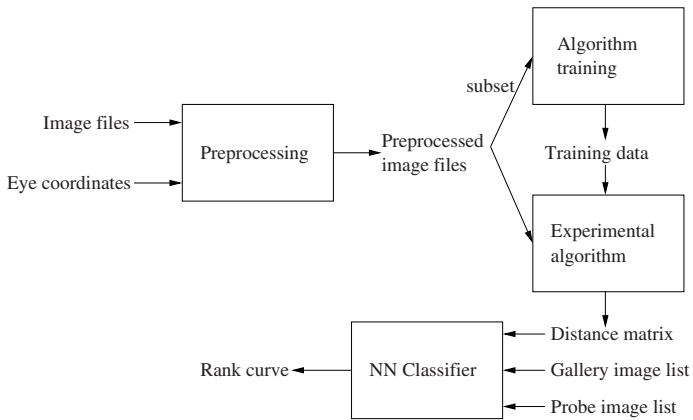


Fig. 3. The parts of the CSU face recognition system.

3. The preprocessed images are fed into the experimental algorithm which outputs a distance matrix containing the distance between each pair of images.
4. Using the distance matrix and different settings for gallery and probe image sets, the system calculates rank curves for the system. These can be calculated for prespecified gallery and probe image sets or by choosing a random permutations of one large set as probe and gallery sets and calculating the average performance. The advantage of the prior method is that it is easy to measure the performance of the algorithm under certain challenges (e.g. different lighting conditions) whereas the latter is more reliable statistically.

The CSU system uses the same gallery and probe image sets that were used in the original FERET test. Each set contains at most one image per person. These sets are:

- **fa** set, used as a gallery set, contains frontal images of 1196 people.
- **fb** set (1195 images). The subjects were asked for an alternative facial expression than in fa photograph.
- **fc** set (194 images). The photos were taken under different lighting conditions.
- **dup I** set (722 images). The photos were taken later in time.
- **dup II** set (234 images). This is a subset of the dup I set containing those images that were taken at least a year after the corresponding gallery image.

In this paper, we use two statistics produced by the permutation tool: the mean recognition rate with a 95 % confidence interval and the probability of one algorithm outperforming another [13]. The image list used by the tool¹ contains 4 images of each of the 160 subjects. One image of every subject is selected to the gallery set and another image to the probe set on each permutation. The number of permutations is 10000.

¹ list640.srt in the CSU Face Identification Evaluation System package

The CSU system comes with implementations of the PCA, LDA, Bayesian intra/extrapersonal (BIC) and Elastic Bunch Graph Matching (EBGM) face recognition algorithms. We include the results obtained with PCA, BIC² and EBGM here for comparison.

There are some parameters that can be chosen to optimise the performance of the proposed algorithm. The first one is choosing the LBP operator. Choosing an operator that produces a large amount of different labels makes the histogram long and thus calculating the distance matrix gets slow. Using a small number of labels makes the feature vector shorter but also means losing more information. A small radius of the operator makes the information encoded in the histogram more local. The number of labels for a neighbourhood of 8 pixels is 256 for standard LBP and 59 for LBP^{u2}. For the 16-neighbourhood the numbers are 65536 and 243, respectively. The usage of uniform patterns is motivated by the fact that most patterns in facial images are uniform: we found out that in the preprocessed FERET images, 79.3 % of all the patterns produced by the LBP_{16,2} operator are uniform.

Another parameter is the division of the images into regions R_0, \dots, R_{m-1} . The length of the feature vector becomes $B = mB_r$, in which m is the number of regions and B_r is the LBP histogram length. A large number of small regions produces long feature vectors causing high memory consumption and slow classification, whereas using large regions causes more spatial information to be lost. We chose to divide the image with a grid into $k * k$ equally sized rectangular regions (windows). See Figure 5 (a) for an example of a preprocessed facial image divided into 49 windows.

4 Results

To assess the performance of the three proposed distance measures, we chose to use two different LBP operators in windows of varying size. We calculated the distance matrices for each of the different settings and used the permutation tool to calculate the probabilities of the measures outperforming each other. The results are in Table 1.

From the statistical hypothesis testing point of view, it cannot be said that any of the metrics would be the best one with a high (>0.95) probability. However, histogram intersection and χ^2 measures are clearly better than log-likelihood when the average number of labels per histogram bin is low but log-likelihood performs better when this number increases. The log-likelihood measure has been preferred for texture images [8] but because of its poor performance on small windows in our experiments it is not appealing for face recognition. The χ^2 measure performs slightly better than histogram intersection so we chose to use it despite the simplicity of the histogram intersection.

When looking for the optimal window size and LBP operator we noticed that the LBP representation is quite robust with respect to the selection of the

² Two decision rules can be used with the BIC classifier: Maximum A Posteriori (MAP) or Maximum Likelihood (ML). We include here the results obtained with MAP.

Table 1. The performance of the histogram intersection, log-likelihood and χ^2 dissimilarity measures using different window sizes and LBP operators.

Operator	Window size	$P(\text{HI} > \text{LL})$	$P(\chi^2 > \text{HI})$	$P(\chi^2 > \text{LL})$
$\text{LBP}_{8,1}^{u2}$	18x21	1.000	0.714	1.000
$\text{LBP}_{8,1}^{u2}$	21x25	1.000	0.609	1.000
$\text{LBP}_{8,1}^{u2}$	26x30	0.309	0.806	0.587
$\text{LBP}_{16,2}^{u2}$	18x21	1.000	0.850	1.000
$\text{LBP}_{16,2}^{u2}$	21x25	1.000	0.874	1.000
$\text{LBP}_{16,2}^{u2}$	26x30	1.000	0.918	1.000
$\text{LBP}_{16,2}^{u2}$	32x37	1.000	0.933	1.000
$\text{LBP}_{16,2}^{u2}$	43x50	0.085	0.897	0.418

parameters. Changes in the parameters may cause big differences in the length of the feature vector, but the overall performance is not necessarily affected significantly. For example, changing from $\text{LBP}_{16,2}^{u2}$ in 18*21-sized windows to $\text{LBP}_{8,2}^{u2}$ in 21*25-sized windows drops the histogram length from 11907 to 2124, while the mean recognition rate reduces from 76.9 % to 73.8 %.

The mean recognition rates for the $\text{LBP}_{16,2}^{u2}$, $\text{LBP}_{8,2}^{u2}$ and $\text{LBP}_{8,1}^{u2}$ as a function of the window size are plotted in Figure 4. The original 130*150 pixel image was divided into $k * k$ windows, $k = 4, 5, \dots, 11, 13, 16$ resulting in window sizes from 32*37 to 8*9. The five smallest windows were not tested using the $\text{LBP}_{16,2}^{u2}$ operator because of the high dimension of the feature vector that would have been produced. As expected, a larger window size induces a decreased recognition rate because of the loss of spatial information. The $\text{LBP}_{8,2}^{u2}$ operator in 18*21 pixel windows was selected since it is a good trade-off between recognition performance and feature vector length.

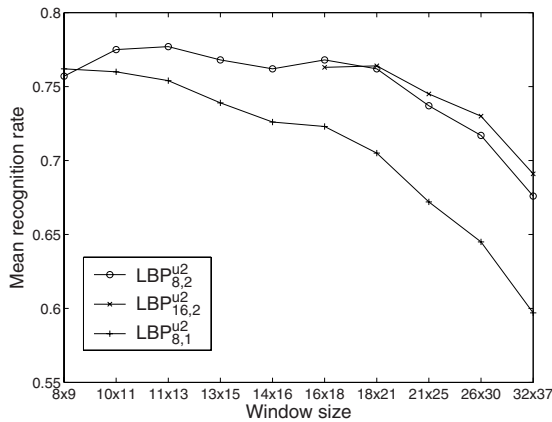


Fig. 4. The mean recognition rate for three LBP operators as a function of the window size.

To find the weights w_j for the weighted χ^2 statistic (Equation 6), the following procedure was adopted: a training set was classified using only one of the 18*21 windows at a time. The recognition rates of corresponding windows on the left and right half of the face were averaged. Then the windows whose rate lay below the 0.2 percentile of the rates got weight 0 and windows whose rate lay above the 0.8 and 0.9 percentile got weights 2.0 and 4.0, respectively. The other windows got weight 1.0.

The CSU system comes with two training sets, the standard FERET training set and the CSU training set. As shown in Table 2, these sets are basically subsets of the *fa*, *fb* and *dup I* sets. Since illumination changes pose a major challenge to most face recognition algorithms and none of the images in the *fc* set were included in the standard training sets, we defined a third training set, called the subfc training set, which contains half of the *fc* set (subjects 1013–1109).

Table 2. Number of images in common between different training and testing sets.

Training set	fa	fb	fc	dup I	dup II	Total number of images
FERET standard	270	270	0	184	0	736
CSU standard	396	0	0	99	0	501
subfc	97	0	97	0	0	194

The permutation tool was used to compare the weights computed from the different training sets. The weights obtained using the FERET standard set gave an average recognition rate of 0.80, the CSU standard set 0.78 and the subfc set 0.81. The pairwise comparison showed that the weights obtained with the subfc set are likely to be better than the others ($P(\text{subfc} > \text{FERET})=0.66$ and $P(\text{subfc} > \text{CSU})=0.88$).

The weights computed using the subfc set are illustrated in Figure 5 (b). The weights were selected without utilising an actual optimisation procedure and thus they are probably not optimal. Despite that, in comparison with the nonweighted method, we got an improvement both in the processing time (see Table 3) and recognition rate ($P(\text{weighted} > \text{nonweighted})=0.976$).

The image set which was used to determine the weights overlaps with the *fc* set. To avoid biased results, we preserved the other half of the *fc* set (subjects

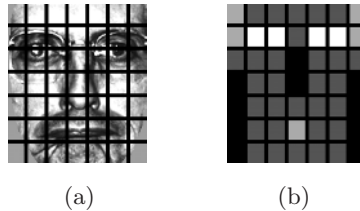


Fig. 5. (a) An example of a facial image divided into 7x7 windows. (b) The weights set for weighted χ^2 dissimilarity measure. Black squares indicate weight 0.0, dark grey 1.0, light grey 2.0 and white 4.0.

Table 3. Processing times of weighted and nonweighted LBP on a 1800 MHz AMD Athlon running Linux. Note that processing FERET images (last column) includes heavy disk operations, most notably writing the distance matrix of about 400 MB to disk.

Type of LBP	Feature ext.	Distance calc.	Processing
	(ms / image)	(μ s / pair)	FERET images (s)
Weighted	3.49	46.6	1046
Nonweighted	4.14	58.6	1285

1110-1206) as a validation set. Introducing the weights increased the recognition rate for the training set from 0.49 to 0.81 and for the validation set from 0.52 to 0.77. The improvement is slightly higher for the training set, but the significant improvement for the validation set implies that the calculated weights generalize well outside the training set.

The final recognition results for the proposed method are in shown Table 4 and the rank curves are plotted in Figures 6 (a)–(d). LBP clearly outperforms the control algorithms in all the FERET test sets and in the statistical test. It should be noted that the CSU implementations of the algorithms whose results we included here do not achieve the same figures as in the original FERET test due to some modifications in the experimental setup as mentioned in [11]. The results of the original FERET test can be found in [12].

Table 4. The recognition rates of the LBP and comparison algorithms for the FERET probe sets and the mean recognition rate of the permutation test with a 95 % confidence interval.

Method	fb	fc	dup I	dup II	lower	mean	upper
LBP, weighted	0.97	0.79	0.66	0.64	0.76	0.81	0.85
LBP, nonweighted	0.93	0.51	0.61	0.50	0.71	0.76	0.81
PCA, MahCosine	0.85	0.65	0.44	0.22	0.66	0.72	0.78
Bayesian, MAP	0.82	0.37	0.52	0.32	0.67	0.72	0.78
EBGM_Optimal	0.90	0.42	0.46	0.24	0.61	0.66	0.71

Additionally, to gain knowledge about the robustness of our method against slight variations of pose angle and alignment we tested our approach on the ORL face database (Olivetti Research Laboratory, Cambridge) [14]. The database contains 10 different images of 40 distinct subjects (individuals). Some images were taken at different times for some people. There are variations in facial expression (open/closed eyes, smiling/non-smiling.), facial details (glasses/no glasses) and scale (variation of up to about 10 %). All the images were taken against a dark homogenous background with the subjects in an upright, frontal position, with tolerance for some tilting and rotation of up to about 20 degrees. The images are grey scale with a resolution of 92*112. Randomly selecting 5 images for the gallery set and the other 5 for the probe set, the preliminary experiments result in 0.98 of average recognition rate and 0.012 of standard

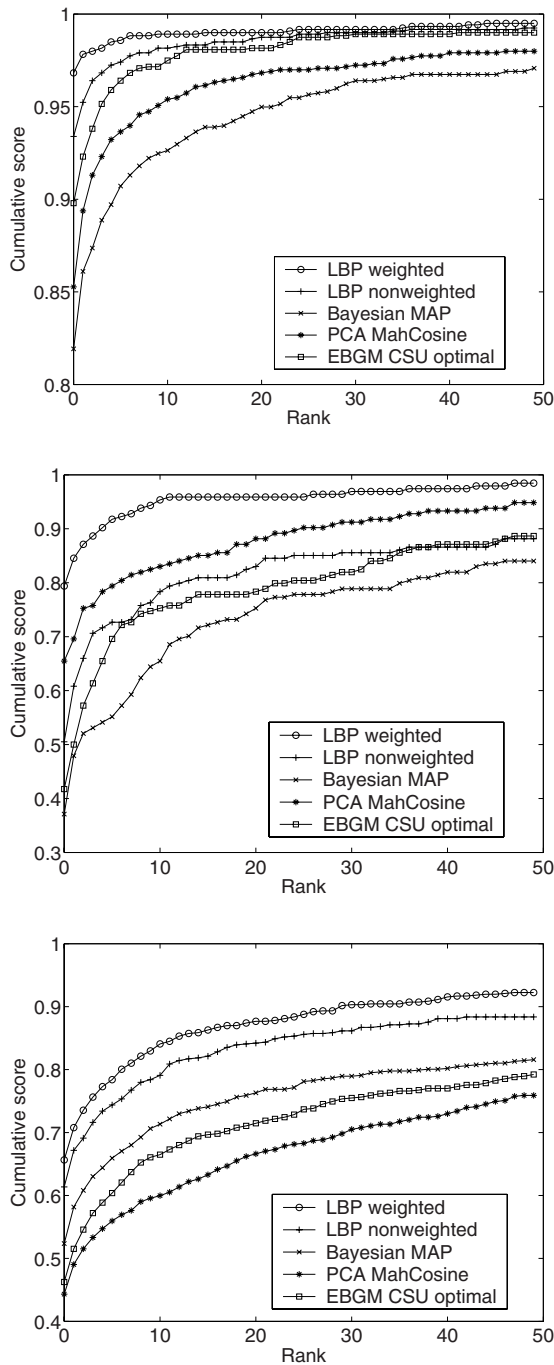


Fig. 6. (a), (b), (c) Rank curves for the *fb*, *fc* and *dup1* probe sets (from top to down).

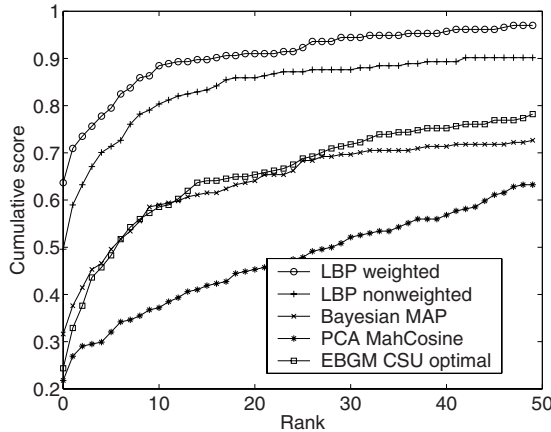


Fig. 6. (d) Rank curve for the *dup2* probe set.

deviation of 100 random permutations using $\text{LBP}_{16,2}^{u2}$, a windows size of 30×37 and χ^2 as a dissimilarity measure. Window weights were not used. Note that no registration or preprocessing was made on the images. The good results indicate that our approach is also relatively robust with respect to alignment. However, because of the lack of a standardised protocol for evaluating and comparing systems on the ORL database, it is too difficult to include here a fair comparison with other approaches that have been tested using ORL.

5 Discussion and Conclusion

Face images can be seen as a composition of micro-patterns which can be well described by LBP. We exploited this observation and proposed a simple and efficient representation for face recognition. In our approach, a face image is first divided into several blocks (facial regions) from which we extract local binary patterns and construct a global feature histogram that represents both the statistics of the facial micro-patterns and their spatial locations. Then, face recognition is performed using a nearest neighbour classifier in the computed feature space with χ^2 as a dissimilarity measure. The proposed face representation can be easily extracted in a single scan through the image, without any complex analysis as in the EBGM algorithm.

We implemented the proposed approach and compared it against well-known methods such as PCA, EBGM and BIC. To achieve a fair comparison, we considered the FERET face database and protocol, which are a *de facto* standard in face recognition research. In addition, we adopted normalisation steps and implementation of the different algorithms (PCA, EBGM and BIC) from the CSU face identification evaluation system. Reporting our results in such a way does not only make the comparative study fair but also offers the research community new performances to which they are invited to compare their results.

The experimental results clearly show that the LBP-based method outperforms other approaches on all probe sets (*fb*, *fc*, *dup I* and *dup II*). For instance, our method achieved a recognition rate of 97% in the case of recognising faces under different facial expressions (*fb* set), while the best performance among the tested methods did not exceed 90%. Under different lighting conditions (*fc* set), the LBP-based approach has also achieved the best performance with a recognition rate of 79% against 65%, 37% and 42% for PCA, BIC and EBGM, respectively. The relatively poor results on the *fc* set confirm that illumination change is still a challenge to face recognition. Additionally, recognising duplicate faces (when the photos are taken later in time) is another challenge, although our proposed method performed better than the others.

To assess the performance of the LBP-based method on different datasets, we also considered the ORL face database. The experiments not only confirmed the validity of our approach, but also demonstrated its relative robustness against changes in alignment.

Analyzing the different parameters in extracting the face representation, we noticed a relative insensitivity to the choice of the LBP operator and region size. This is an interesting result since the other considered approaches are more sensitive to their free parameters. This means that only simple calculations are needed for the LBP description while some other methods use exhaustive training to find their optimal parameters.

In deriving the face representation, we divided the face image into several regions. We used only rectangular regions each of the same size but other divisions are also possible as regions of different sizes and shapes could be used. To improve our system, we analyzed the importance of each region. This is motivated by the psychophysical findings which indicate that some facial features (such as eyes) play more important roles in face recognition than other features (such as the nose). Thus we calculated and assigned weights from 0 to 4 to the regions (See Figure 5 (b)). Although this kind of simple approach was adopted to compute the weights, improvements were still obtained. We are currently investigating approaches for dividing the image into regions and finding more optimal weights for them.

Although we clearly showed the simplicity of LBP-based face representation extraction and its robustness with respect to facial expression, aging, illumination and alignment, some improvements are still possible. For instance, one drawback of our approach lies in the length of the feature vector which is used for face representation. Indeed, using a feature vector length of 2301 slows down the recognition speed especially, for very large face databases. A possible direction is to apply a dimensionality reduction to the face feature vectors. However, due to the good results we have obtained, we expect that the methodology presented here is applicable to several other object recognition tasks as well.

Acknowledgements. This research was supported in part by the Academy of Finland.

References

1. Phillips, P., Grother, P., Micheals, R.J., Blackburn, D.M., Tabassi, E., Bone, J.M.: Face recognition vendor test 2002 results. Technical report (2003)
2. Zhao, W., Chellappa, R., Rosenfeld, A., Phillips, P.J.: Face recognition: a literature survey. Technical Report CAR-TR-948, Center for Automation Research, University of Maryland (2002)
3. Phillips, P.J., Wechsler, H., Huang, J., Rauss, P.: The FERET database and evaluation procedure for face recognition algorithms. *Image and Vision Computing* **16** (1998) 295–306
4. Turk, M., Pentland, A.: Eigenfaces for recognition. *Journal of Cognitive Neuroscience* **3** (1991) 71–86
5. Etemad, K., Chellappa, R.: Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America* **14** (1997) 1724–1733
6. Wiskott, L., Fellous, J.M., Kuiger, N., von der Malsburg, C.: Face recognition by elastic bunch graph matching. *IEEE Transaction on Pattern Analysis and Machine Intelligence* **19** (1997) 775–779
7. Moghaddam, B., Nastar, C., Pentland, A.: A bayesian similarity measure for direct image matching. In: 13th International Conference on Pattern Recognition. (1996) II: 350–358
8. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (2002) 971–987
9. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition* **29** (1996) 51–59
10. Gong, S., McKenna, S.J., Psarrou, A.: *Dynamic Vision, From Images to Face Recognition*. Imperial College Press, London (2000)
11. Bolme, D.S., Beveridge, J.R., Teixeira, M., Draper, B.A.: The CSU face identification evaluation system: Its purpose, features and structure. In: Third International Conference on Computer Vision Systems. (2003) 304–311
12. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** (2000) 1090–1104
13. Beveridge, J.R., She, K., Draper, B.A., Givens, G.H.: A nonparametric statistical comparison of principal component and linear discriminant subspaces for face recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2001) I: 535–542
14. Samaria, F.S., Harter, A.C.: Parameterisation of a stochastic model for human face identification. In: IEEE Workshop on Applications of Computer Vision. (1994) 138–142

Steering in Scale Space to Optimally Detect Image Structures

Jeffrey Ng and Anil A. Bharath

Faculty of Engineering,
Imperial College London,
United Kingdom SW7 2AZ
{jeffrey.ng,a.bharath}@imperial.ac.uk

Abstract. Detecting low-level image features such as edges and ridges with spatial filters is improved if the scale of the features are known *a priori*. Scale-space representations and wavelet pyramids address the problem by using filters over multiple scales. However, the scales of the filters are still fixed beforehand and the number of scales is limited by computational power. The filtering operations are thus not adapted to detect image structures at their optimal or intrinsic scales. We adopt the steering approach to obtain filter responses at arbitrary scales from a small set of filters at scales chosen to accurately sample the “scale space” within a given range. In particular, we use the Moore-Penrose inverse to learn the steering coefficients, which we then regress by polynomial function fitting to the scale parameter in order to steer the filter responses continuously across scales. We show that the extrema of the polynomial steering functions can be easily computed to detect interesting features such as phase-independent energy maxima. Such points of energy maxima in our α -scale-space correspond to the intrinsic scale of the filtered image structures. We apply the technique to several well-known images to segment image structures which are mostly characterised by their intrinsic scale.

1 Introduction

Low-level feature detection and extraction by spatial filters is used in many fields such as image analysis, image representation, image compression and computer vision. In applications where the aim of the spatial filtering is to reconstruct the original image, e.g. compression, the choice of filter size and scale affect the spatial frequency of the encoded and reconstructed image structures. A set of filters at different scales can be employed to encode coarse, medium and fine structures in the image [1]. However, an image structure may contribute significantly to multiple filter responses over different scales when the filter scale does not match the scale of the image structure. While this does not affect the reconstruction process, inferring the presence of low-level features from filter responses becomes difficult because of the ambiguity arising from the sub-optimal encoding of image structures and the possible encoding of multiple structures into a single filter response when incorrect filter scales are used.

The problem of incorrect filter scale has previously been addressed by filtering over a greater number of finely-sampled scales. Fdez-Valdivia [2] constructed a bank of 2D Gabor filters over multiple orientations and scales and used the normalised 2D power spectrum over the filter responses to detect activation at scales where image structures exist. Lindeberg [3] uses extrema over scales in normalised scale-space to optimally detect image structures at their intrinsic scale. Kadir and Brady [4] have shown that normalised maxima of entropy of low-level features in images can be used to detect salient image structures. There is also evidence that neurons in the primary visual cortex of Old World monkeys “tune” their spatial frequency response over time to detect features from coarse to fine scale [5]. This spatial frequency tuning highlights the importance of searching over a range of scales to optimally detect image features.

The main problem with these approaches is that exhaustive filtering with kernels over a wide range of finely-sampled scales is computationally intensive and inefficient. Fast implementations of scale selection by searching for scale-space maxima has been proposed by [6,7] using quadratic interpolation of the filter responses across scales. On the other hand, Freeman and Adelson [8] devised an analytical method for linearly combining all the responses of a small set of oriented basis filters to obtain filter responses over all orientations. While linearly combining the basis filter responses over orientations results in a Taylor series expansion, an analytical formulation for steering filter responses across the scale parameter is not so forthcoming because of the unbounded nature of the problem, i.e. filter scales can theoretically increase to infinity. In practice, the maximum filter scale is limited by the size of the image. Perona [9] used the Singular Value Decomposition to design scale-steerable “deformable” filters. Bharath [10] constructed exemplar vectors of the radial frequency response of filters across scales and used a Moore-Penrose generalised inverse to learn the steering coefficients through simple matrix algebra. We modify the technique by learning the steering coefficients from radial response exemplars in the spatial domain and we parameterise the coefficients in terms of scale through regressive fitting of polynomial functions.

In Section 2, we describe a method for specifying the angular and radial frequency characteristics of filter kernels in the Fourier domain to construct an appropriate scale-space and then generating filter mask coefficients in the spatial domain by the inverse discrete Fourier transform. In Section 3, we exploit the ability to synthesise filters of arbitrary radial frequency responses and thus spatial scale in order to build two exemplar matrices containing desired radial responses in the spatial domain over a finely-sampled range of scales and over a small fixed set of basis scales respectively. We use the Moore-Penrose inverse to learn the linear combination coefficients to compute the former from the latter. The linear combination coefficients are then parameterised over scale by polynomial functions, yielding continuous scale steering functions. After filtering an image, we collapse in Section 4 the fields of filter responses of the basis set and the individual polynomial steering functions for each basis filter into a two-dimensional field of single polynomial steering functions. The maxima of these single polynomial steering functions can be easily obtained by computing

the roots of their derivative. Although we cannot directly steer the energy of a pair of complex filters in quadrature using this method, we can still compute the location of the maxima of the square of the energy which is the same as the location of the maxima of the energy. We show that such energy maxima correspond to the intrinsic scale of the image structure being filtered in a similar manner to [2,3,4]. We finally show how the scale at which energy maxima occurs in an image can be used to segment a class of well-known natural images where intrinsic scale plays an important role in defining structures in the image.

2 Design of Polar Separable Filters

Characteristics of local low-level image structures at any position consist of orientation, scale and phase [9]. In order to detect such structures, spatial filters are designed with similar orientation, scale and phase characteristics. To steer through the scale parameter, we need to design filters where the other parameters are kept constant. Polar separable filter kernels allow the radial frequency (scale) characteristic of the filters to be separately specified from the angular frequency (orientation) characteristic [11]. Furthermore, filtering with pairs of filter kernels in quadrature, where one filter is the Hilbert transform of the other, yields a phase-independent “energy” response.

The polar separable filter kernel $G_\alpha^\theta(\omega, \phi)$ is specified in the Fourier domain by the radial frequency function $G_\alpha(\omega)$ and angular frequency function $G^\theta(\phi)$ where α and θ are the scale and orientation of the desired filter respectively

$$G_\alpha^\theta(\omega, \phi) = G_\alpha(\omega)G^\theta(\phi) \quad (1)$$

The angular frequency characteristic of the filter affects the selectivity of its response to a specific range of orientations of image structures. The angular power (sum of squares) of a set of oriented filters for covering orientations $[0, \pi]$ also needs to be flat in order to provide uniform coverage. We choose a third power cosine function, clipped by a rectangular function, which gives a flat angular power response when used in a set of four orientations, i.e. $\theta \in \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$

$$G^\theta(\phi) = \cos^3(\phi - \theta) \text{rect}(\phi - \theta) \quad (2)$$

where *rect* is the unit rectangular function

$$\text{rect}(\phi) = \begin{cases} 1, & \text{if } |\phi| \leq \frac{\pi}{2} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

The radial frequency characteristic of the filter affects the spread of its power spectra over scales. Traditional linear scale-space representations use derivatives of Gaussian in the spatial domain [12]. However, the amplitude of Gaussian spatial derivatives decrease over scales for scaled versions of the same image structure, rendering comparison of filter responses across scales more complex. Lindeberg [3] thus introduces an L_p normalisation of the scale-space filter responses so that their maxima corresponds to the optimal detection of image

structures at their intrinsic scale. On the other hand, Poisson kernels have recently been investigated for the consistency of filter behaviour over scales [13]. We adopt Erlang functions [14] of order $n = 7$ and scale α , which possess similar radial frequency responses to Poisson kernels and also benefit from quasi-invariant energy response over scales α avoiding the need for scale normalisation (which we do not show here due to lack of space)

$$G_\alpha(\omega) = \left(\frac{\alpha e}{n}\right)^n \omega^n e^{-\alpha\omega} \quad (4)$$

The inverse discrete Fourier transform of $G_\alpha^\theta(\omega, \phi)$ yields a complex filter $g_\alpha^\theta(\omega, \phi)$ in the spatial domain where the real and imaginary parts of the kernel are in quadrature. The magnitude of the response of this complex filter, when convolved with an image, provides a phase-independent “energy” response [11]. Following this approach, complex filter kernels of arbitrary orientation and scale can be synthesised.

3 Steering in Scale

A family of spatial filter kernels with similar orientation but varying scales can be obtained by varying the α scale parameter in Eqn (4). A scale-space representation similar to [1] can easily be constructed, providing a continuum of filter responses across scales. However, filtering at very small scale intervals to obtain a continuum of filter responses is a computationally intensive operation. Perona [9] approached the problem of designing a steerable basis filter set by first choosing the number of filters and thus the number of filtering operations. Then, Singular Value Decomposition is used to synthesise the best filter kernels for a certain detection task such as steering across scales. In contrast, we start with polar separable filter kernels where the desired orientation and radial characteristics are pre-specified and use the method of Bharath [15] to find the steering coefficients. Bharath generated radial frequency responses across many scales at very fine intervals and used a Moore-Penrose generalised inverse to learn the steering coefficients for obtaining those responses from the linear combination of a small set of basis filters. He varied both the scale parameter α and the order n of the Erlang functions in order to construct his basis set.

To create steerable filters, we first assume that the steering of filter responses across scales can be obtained by linearly combining the responses $f_{\alpha_i}^\theta(x, y)$, where $i \in \{1, \dots, N\}$, of a small set of N basis filters $G_{\alpha_i}^\theta(x, y)$ as

$$f_\alpha^\theta(x, y) = \sum_{i=1}^N s_{i,\alpha} f_{\alpha_i}^\theta(x, y) \quad (5)$$

where $s_{i,\alpha}$ is the steering coefficient of filter i for scale α . We can also formulate Eqn (5) into matrix form

$$\mathbf{F} = \mathbf{F}_\mathbf{B} \mathbf{S} \quad (6)$$

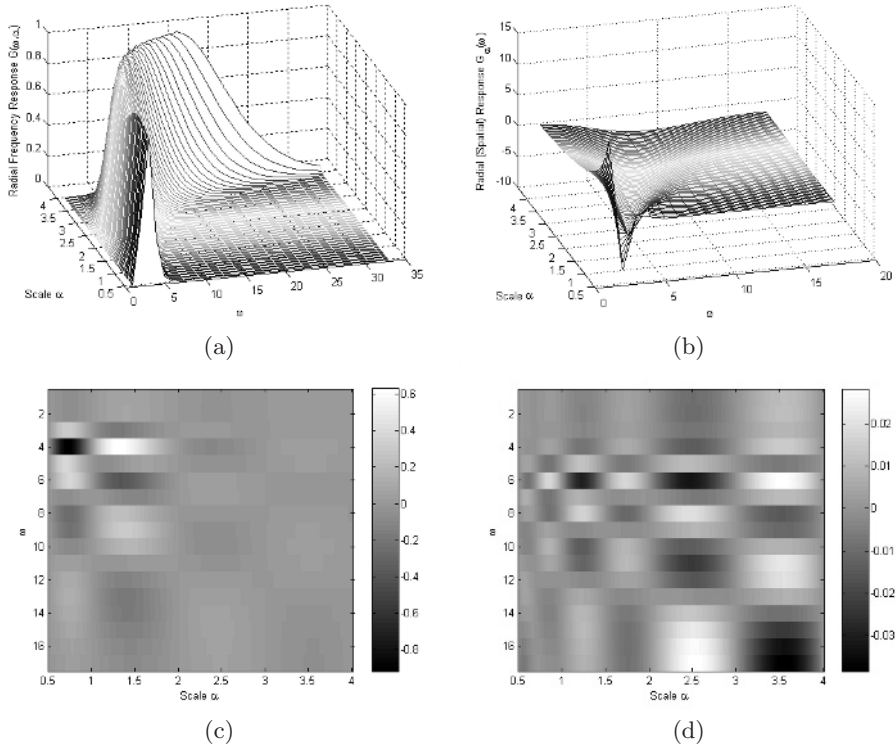


Fig. 1. From left to right: (a) Radial frequency response in the Fourier domain over scales α ; (b) Real part of the radial response in the spatial domain over scales; (c) Error in the real part of the steered radial response from basis filters at scales 0.5, 1.0, 2.0, 3.0, 4.0; (d) Error in the real part of the steered radial response from basis filters at scales 0.5, 0.7, 1.0, 1.5, 2.0, 3.0, 4.0. Please note that the range of radial responses in (b) is $[-7.69, 13.31]$.

where \mathbf{F} is a matrix of column vectors of the radial responses ($G_\alpha(\omega)$ in the spatial domain) across scales $[\alpha_1, \alpha_N]$ at very small intervals, \mathbf{F}_B is a matrix of column vectors of radial responses from the basis set with scales $\{\alpha_1, \dots, \alpha_N\}$ and \mathbf{S} is a matrix of column vectors of steering coefficients to obtain each column of \mathbf{F} from a linear combination of \mathbf{F}_B . Given that we can synthesise the matrix \mathbf{F} by varying the scale parameter α in Eqn (4) in very small increments between $[\alpha_1, \alpha_N]$ and we can also synthesise \mathbf{F}_B , we obtain \mathbf{S} by the Moore-Penrose generalised inverse

$$\mathbf{S} = \mathbf{F}_B^\dagger \mathbf{F} \quad (7)$$

where

$$\mathbf{B}^\dagger = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \quad (8)$$

We can also evaluate the accuracy of the learnt steering solution for a given basis filter set by re-evaluating the steered radial responses

$$\hat{\mathbf{F}} = \mathbf{F}_B \mathbf{S} \quad (9)$$

and thus empirically choose the scales α_i of the basis filter set to obtain an acceptable steering accuracy.

In order to select the scales of our basis filter set, we observe that the radial frequency response $G_\alpha(\omega)$ changes smoothly with the scale parameter α as shown in Fig. 1(a). The change in radial response in the frequency domain increases exponentially with α . In the spatial domain (Fig. 1(b)), this translates into an inverse exponential rate of change for the real part of the radial response, where most of the change occurs at small values of α . Therefore, we dedicate more basis filters to the finer scales than the coarser scales. In Fig. 1(c), we show the steering error $\hat{\mathbf{F}} - \mathbf{F}$ resulting from filter scales chosen at integer intervals (except the finest scale to avoid any aliasing), i.e. scales $\alpha_i \in \{0.5, 1.0, 2.0, 3.0, 4.0\}$. The steering errors occur mostly at the finer scales because of the scaling properties of the Erlang function. As we dedicate more basis filters to the finer scales, i.e. $\alpha_i \in \{0.5, 0.75, 1.0, 1.5, 2.0, 3.0, 4.0\}$, the steering errors are spread out more evenly and the magnitude of the errors is significantly reduced. We henceforth use the latter set of scales for our basis filters. The size of the filter kernels in the spatial domain range from 9×9 to 60×60 . Fig. 1 suggests that the range of scales over which the filter responses can be steered can be efficiently increased by adding coarser-scale basis filters over larger intervals of α .

Each N -element column of \mathbf{S} in Eqn (7) contains the steering coefficients for the N basis filters to obtain filter responses at the corresponding scale α . Therefore, each of the N rows of \mathbf{S} contains the finely sampled steering coefficients across scales for the N basis filters. We fit 12^{th} order polynomial functions to regress the steering coefficients for each basis filter to the scale parameter α . We thus replace the steering coefficients $s_{i,\alpha}$ in Eqn (5) by polynomial steering functions¹ $s_i(\alpha)$ with polynomial coefficients cf_i^p where $p \in \{0, \dots, 12\}$

$$s_i(\alpha) = cf_i^{12} \alpha^{12} + \dots + cf_i^1 \alpha + cf_i^0 \quad (10)$$

to obtain

$$f_\alpha^\theta(x, y) = \sum_{i=1}^N s_i(\alpha) f_{\alpha_i}^\theta(x, y) \quad (11)$$

We show scale steering results on Jaehne's test image of radial cosine modulation with increasing frequency from the centre in Fig. 2. In order to show the accuracy of the complex steered responses, we show the energy (defined below in Eqn (16)) of the responses, which is independent of the phase of the cosine modulation, for orientation $\theta = 0$.

¹ We refer to the polynomial functions which provide the steering coefficients for a given basis filter from a scale parameter *alpha* as a *steering function*

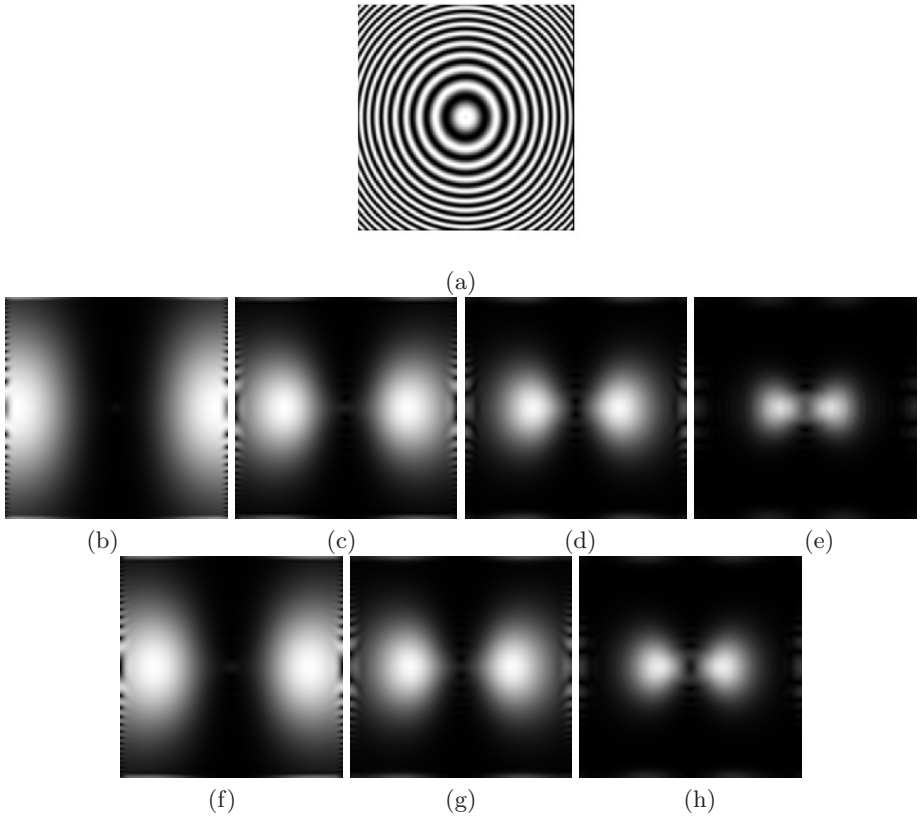


Fig. 2. From left to right: (a) Jaehne's test image with varying radial frequency; (b)-(e) Energy of the complex basis filter responses at scales 1.0, 1.5, 2 and 3 for orientation $\theta = 0$ (vertical edges); and (f)-(h) Energy of the steered filter responses at intermediate scales 1.25, 1.75 and 2.5 for orientation $\theta = 0$.

4 Detecting Intrinsic Scale

The polynomial steering functions allow one to obtain filter responses at any scale within the range of scales of the basis filter set. Kadir and Brady [4] have shown that normalised maxima of entropy over scales can be used to detect salient regions in images. Lindeberg [3] used normalised extrema in scale-space to detect edges and ridges more reliably. The formulation of the steering functions $s_i(\alpha)$ in terms of polynomial functions of α lends itself well to the detection of global maxima by analytically finding the roots of the derivatives of the polynomials, rather than exhaustively performing operations over multiple scales to detect maxima as in the previous aforementioned works.

Once the complex filter responses $f_\alpha^\theta(x, y)$ have been computed over an image, they can be treated as constants. A polynomial function multiplied by a constant

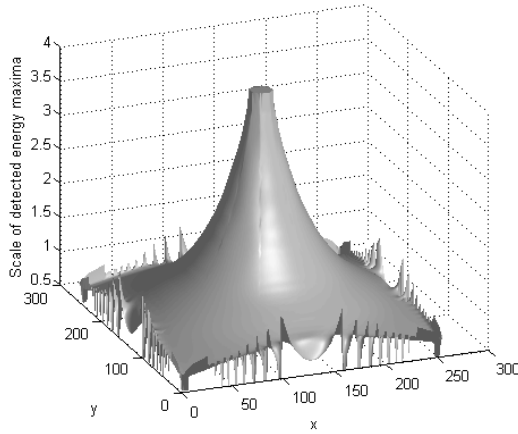


Fig. 3. Scale at which energy maxima were detected in Jaehne's test image, corresponding to the intrinsic scale function used to generate the test pattern. Note: Some noise at the borders remains even though filtering was done with border reflection.

results in a new polynomial function where the coefficients are scaled by the constant. We can thus multiply our polynomial steering functions $s_i(\alpha)$ by our field of filter response constants $f_\alpha^\theta(x, y)$ in Eqn (11) to obtain fields of polynomial steering functions for locations (x, y) in the image

$$f_\alpha^\theta(x, y) = \sum_{i=1}^N s_i^\theta(\alpha, x, y) \quad (12)$$

where

$$s_i^\theta(\alpha, x, y) = [\text{cf}_i^{12} f_\alpha^\theta(x, y)] \alpha^{12} + \dots + [\text{cf}_i^1 f_\alpha^\theta(x, y)] \alpha + [\text{cf}_i^0 f_\alpha^\theta(x, y)] \quad (13)$$

The filter responses over scales (Eqn (12)) can be further simplified by summing the polynomial functions $s_i^\theta(\alpha, x, y)$ (adding their coefficients) together for all basis filters i to obtain a field of single polynomial steering functions

$$f_\alpha^\theta(x, y) = s^\theta(\alpha, x, y) \quad (14)$$

where

$$s^\theta(\alpha, x, y) = \left[\sum_{i=1}^N \text{cf}_i^{12} f_\alpha^\theta(x, y) \right] \alpha^{12} + \dots + \left[\sum_{i=1}^N \text{cf}_i^1 f_\alpha^\theta(x, y) \right] \alpha + \left[\sum_{i=1}^N \text{cf}_i^0 f_\alpha^\theta(x, y) \right] \quad (15)$$

In essence, the weighted summation of a set of polynomial functions of the same variable α for steering across scales (Eqn (11)) can be simplified into a single polynomial function by weighting and summing the polynomial coefficients only

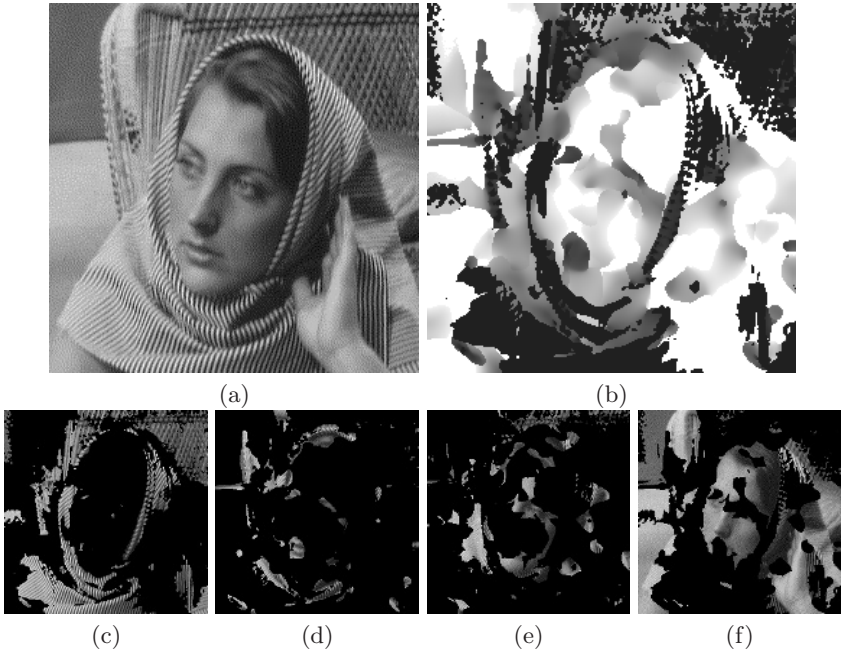


Fig. 4. From left to right, top to bottom: (a) Barbara image; (b) Map of energy maxima over scales (black is 0.5 and white is 4); Segmented Barbara image where energy maxima lies between scales (c) 0.5 and 1, (d) 1 and 2, (e) 2 and 3, and (f) 3 and 4.

(commutation of operations). Finding derivatives of polynomial functions again only involves simple operations on the coefficients.

In order to search for phase-independent maxima of filter responses over scales, whereby ridge-like or edge-like image structures are treated equally, we need to obtain a steering equation for the energy of the complex response. The phase-independent “energy” response of a complex quadrature filter is obtained from the magnitude of the filter response $f_{\alpha}^{\theta}(x, y)$ [11]

$$E_{\alpha}^{\theta}(x, y) = \sqrt{\text{real}(f_{\alpha}^{\theta}(x, y))^2 + \text{imag}(f_{\alpha}^{\theta}(x, y))^2} \quad (16)$$

We can finally collapse the field of complex steering polynomial functions (Eqn (14)) by squaring² the real and imaginary parts of the polynomial steering functions in $f^{\theta}(\alpha, x, y)$ and adding them together. Obtaining the square root of a polynomial function is not a trivial task but fortunately, it is not needed because our final steering function $(E_{\alpha}^{\theta}(x, y))^2$ will have maxima at the same scales as $E_{\alpha}^{\theta}(x, y)$. To obtain an orientation-independent measure of energy maxima over scales, we select the maximum of the energy maxima in each of the four directions $\theta \in \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$. We show the results of the detection of energy maxima

² If the polynomial is represented as a vector of coefficients, convolving the vector with itself yields the square of the polynomial.

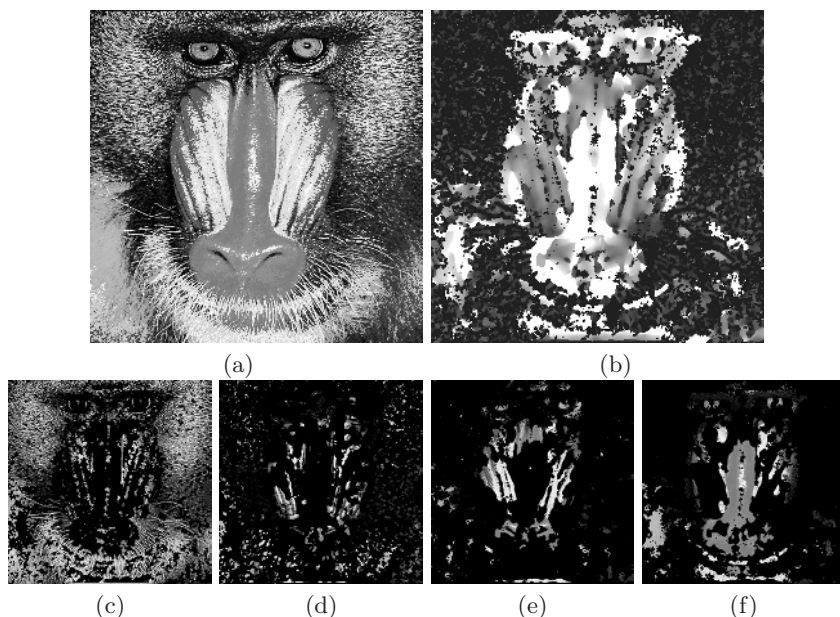


Fig. 5. From left to right, top to bottom: (a) Mandrill image; (b) Map of energy maxima over scales (black is 0.5 and white is 4); Segmented Mandrill image where energy maxima lies between scales (c) 0.5 and 1, (d) 1 and 2, (e) 2 and 3, and (f) 3 and 4.

over scales (and a small set of fixed directions) in Jaehne's test image with varying radial frequency (scale of structures) in Fig. 3. We have used reflection along the borders of the image in order to obtain filtering results near the borders.

5 Experiments

We have applied our scale steering technique to detect maxima of phase-independent energy in popular images, such as Barbara and Mandrill, and other images of natural scenes where intrinsic scale plays an important part in identifying image structures. We show the scales at which energy maxima occur in different parts of the image and show preliminary results in coarsely segmenting the image into four bands: (a) 0.5 – 1.0 for very fine scale structures such as texture, (b) 1.0 – 2.0 for fine scale structures, (c) 2.0 – 3.0 for medium scale structures, and (d) 3.0 – 4.0 for coarse scale structures such as homogenous regions.

In the case of the Barbara image (Fig. 4), the stripes of Barbara's head-scarf and the hatched pattern of the chair in the back have been identified as very fine scale structures, shown in Fig. 4(c). Some folds of the head-scarf, the eyes and the mouth of Barbara have been segmented into the fine scale structures in Fig. 4(d). Barbara's chin and a blurred patch of the striped scarf have been segmented into medium scale structures. Finally, the forehead, the nose, the

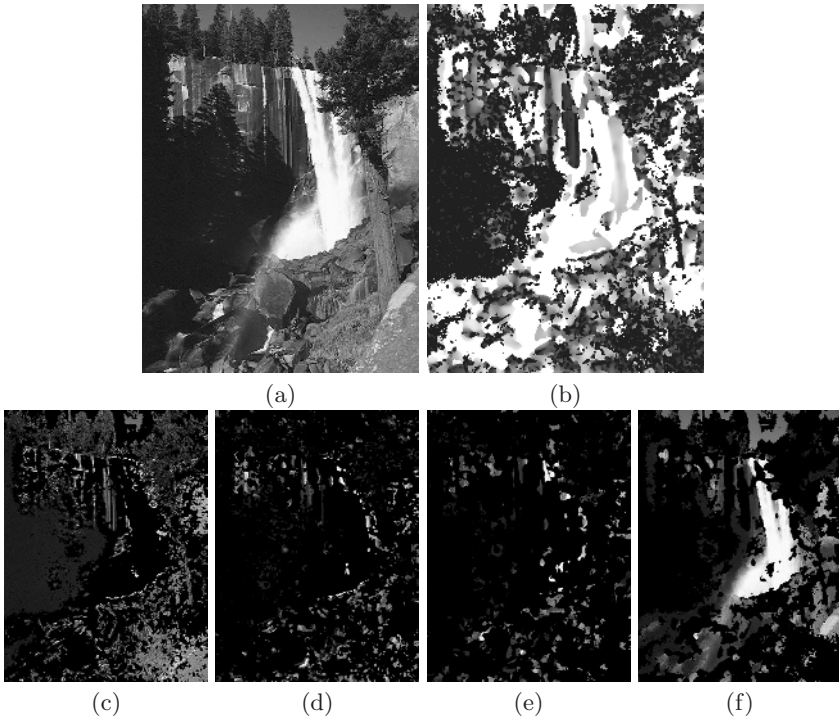


Fig. 6. From left to right, top to bottom: (a) Waterfall image; (b) Map of energy maxima over scales (black is 0.5 and white is 4); Segmented Waterfall image where energy maxima lies between scales (c) 0.5 and 1, (d) 1 and 2, (e) 2 and 3, and (f) 3 and 4.

cheeks, the hands, uniform parts of the wall and floor, and a small piece of scarf and hatched chair were segmented into coarse scale structures in Fig. 4(f).

The ability to segment facial features based on their intrinsic scale can further be seen on the Mandrill image in Fig. 5. The fur, the whiskers and the fine edges are segmented into (c) very fine scale structures. Parts of the nose and some fur are segmented into (d) fine scale structures. Interestingly, the sides of the nose and the eyes are segmented into (e) medium scale structures and the nose is mainly segmented into (f) a coarse scale structure. In Fig. 6, we show how the main waterfall feature is segmented into a (f) coarse scale structure while the trees are segmented into (c) very fine and (d) fine scale structures. Image structures in the sea picture (Fig. 7) are also broken down into (f) coarse structures such as the big waves and the sky, (e) medium structures such as the horizon and some clouds, (c) and (d) for the smaller waves.

6 Conclusion and Future Work

Spatial filters give the highest response when their scale matches that of the image structure being filtered. Filtering over multiple scales such as in scale-space

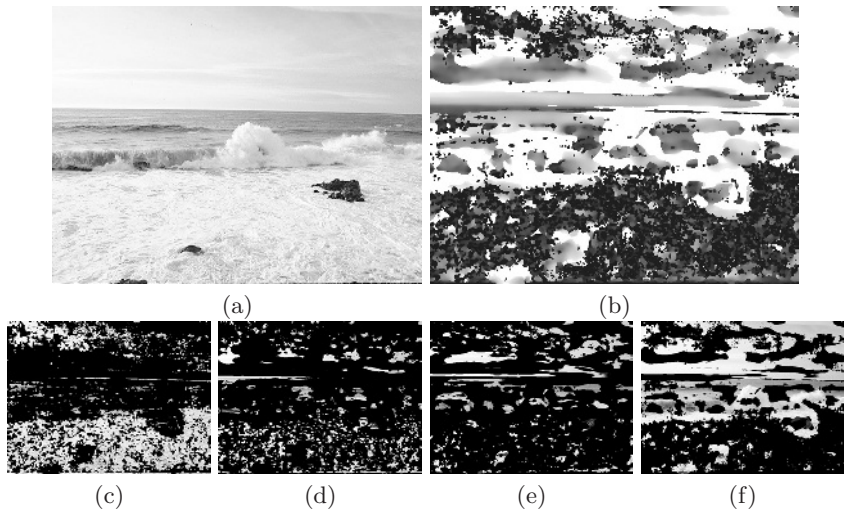


Fig. 7. From left to right, top to bottom: (a) Sea image; (b) Map of energy maxima over scales (black is 0.5 and white is 4); Segmented Sea image where energy maxima lies between scales (c) 0.5 and 1, (d) 1 and 2, (e) 2 and 3, and (f) 3 and 4. Note that the segmentation has separated the clouds which are present in the sky.

representations and pyramids does not fully address the problem as the number of scales is usually constrained by computational power. We have described how to synthesise polar separable filters which possess quasi-invariant responses over scale by adopting Erlang functions as radial frequency characteristic. We have also shown how to create a basis filter set, learn the scale steering coefficients and evaluate the accuracy of the steering solution. The main novelty of our work lies in regressing the scale steering coefficients with polynomial functions and exploiting the ease of collapsing the linear combination of polynomial steering functions into a single polynomial function whose extrema can be analytically computed from the roots of its derivative. We have also shown that maxima of the energy response of our complex filters over scales correspond to the intrinsic scales of image structures both in Jaehne's test image and in natural images.

We have shown results where the global energy maxima over scales were found in each direction first and then the maximum energy over all filtered directions was chosen for determining intrinsic scale. We have not used the remaining roots of the polynomial functions which provide scale information about the other local energy maxima, minima and inflection points occurring both in the direction of greatest energy and the other directions. This information could potentially improve the segmentation results that we provided.

Acknowledgements. This work was funded by the UK Research Council under the Basic Technology Research Programme "Reverse Engineering Human Visual Processes" GR/R87642/02.

References

1. Simoncelli, E., Freeman, W.: The steerable pyramid: A flexible architecture for multi-scale derivative computation. In: IEEE Second International Conference on Image Processing. (1995)
2. Fdez-Valdivia, J., Garcia, J., Martinez-Baena, J., Fdez-Vidal, X.: The selection of natural scales in 2d images using adaptive Gabor filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20** (1998) 458–469
3. Lindeberg, T.: Principles for automatic scale selection. In: *Handbook on Computer Vision and Applications*. Volume 2. Academic Press (1999) 239–274
4. Kadir, T., Brady, M.: Scale, saliency and image description. *International Journal of Computer Vision* **45** (2001) 83–105
5. Bredfeldt, C., Ringach, D.: Dynamics of spatial frequency tuning in macaque v1. *Journal of Neuroscience* **22** (2002) 1976–1984
6. Lindeberg, T., Bretzner, L.: Real-time scale selection in hybrid multi-scale representations. In: *Scale Space 2003*. Volume 2695 of *Lecture Notes in Computer Science*. (2003)
7. Crowley, J., Riff, O.: Fast computation of scale normalised gaussian receptive fields. In: *Scale Space 2003*. Volume 2695 of *Lecture Notes in Computer Science*. (2003) 584–598
8. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13** (1991) 891–906
9. Perona, P.: Deformable kernels for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17** (1995) 488–499
10. Bharath, A.A.: Scale steerability of a class of digital filters. *Electronics Letters* **34** (1998) 1087–1088
11. Granlund, G., Knutsson, H.: *Signal Processing for Computer Vision*. Kluwer Academic Publishers (1994)
12. Witkin, A., Terzopoulos, D., Kass, M.: Signal matching through scale space. *International Journal of Computer Vision* **1** (1987) 133–144
13. Duits, R., Felsberg, M., Florack, L., Platel, B.: α scale spaces on a bounded domain. In: *Scale-Space 2003*. Volume 2695 of *LNCS*. (2003) 494–510
14. Papoulis, A.: *Probability, Random Variables and Stochastic Processes*. McGraw-Hill (1984)
15. Bharath, A.: Steerable filters from Erlang functions. In: *British Machine Vision Conference*. Volume 1. (1998) 144–153

Hand Motion from 3D Point Trajectories and a Smooth Surface Model

Guillaume Dewaele, Frédéric Devernay, and Radu Horaud

INRIA Rhône-Alpes, 38334 Saint Ismier Cedex, France,
<http://www.inrialpes.fr/movi>

Abstract. A method is proposed to track the full hand motion from 3D points reconstructed using a stereoscopic set of cameras. This approach combines the advantages of methods that use 2D motion (e.g. optical flow), and those that use a 3D reconstruction at each time frame to capture the hand motion. Matching either contours or a 3D reconstruction against a 3D hand model is usually very difficult due to self-occlusions and the locally-cylindrical structure of each phalanx in the model, but our use of 3D point trajectories constrains the motion and overcomes these problems.

Our tracking procedure uses both the 3D point matches between two time frames and a smooth surface model of the hand, build with implicit surface. We used animation techniques to represent faithfully the skin motion, especially near joints. Robustness is obtained by using an EM version of the ICP algorithm for matching points between consecutive frames, and the tracked points are then registered to the surface of the hand model. Results are presented on a stereoscopic sequence of a moving hand, and are evaluated using a side view of the sequence.

1 Introduction

Vision-based human motion capture consists in recovering the motion parameters of a human body, such as limb position and joint angles, using camera images. It has a wide range of applications, and has gained much interest lately [12]. Recovering an *unconstrained* hand motion is particularly challenging, mainly because of the high number of degrees of freedom (DOF), the small size of the phalanges, and the multiple self-occlusions that may occur in images of the hand.

One way to obtain hand pose is to compute some appearance descriptors of the hand to extract the corresponding pose from a large database of images [1,18] (*appearance-based methods*), but most approaches work by extracting cues from the images, such as optical flow, contours, or depth, to track the parameters of a 3D hand model (*tracking methods*). Our method belongs to the latter class, and uses data extracted from stereoscopic image sequences, namely trajectories of 3D points, to extract the parameters of a 27 DOF hand model. Other tracking methods may use template images of each phalanx [16], or a combination of optical flow [11,13], contours or silhouette [11,13,20,6,21,9,19], and depth [6]. A few methods are first trying to reduce the number of DOF of the hand model

by analyzing a database of digitized motions [21,9,19], with the risk of losing generality in the set of possible hand poses. In this context multiple cameras were previously used either to recover depth and use it in the tracking process [6], or to recognize the pose in the best view, and then use other images to get a better estimation of the 3D pose [20].

The problem with most previous tracking methods is that using *static cues* (contours or 3D reconstruction), extracted at each time frame, generate a lot of ambiguities in the tracking process, mainly due to the similarity of these cues between fingers, and to the locally-cylindrical structure of fingers which leaves the motion of each individual phalanx unconstrained (although the global hand structure brings the missing constraints). The solution comes from using *motion cues* to bring more constraints. In monocular sequences, the use of optical flow [11] proved to give more stability and robustness to the tracking process, but previous methods using multiple cameras have not yet used 3D motion cues, such as 3D point trajectories, which should improve further the stability of the tracking process. There are three main difficulties in using 3D point trajectories to track the hand pose:

- Since 3D point tracking is a low-level vision process, it generates a number of outliers which the algorithm must be able to detect and discard.
- A 3D point trajectory gives a displacement between two consecutive time frames, but the tracking process may drift with time so that the whole trajectory may not represent the motion of a single physical point.
- Since the skin is a deformable tissue, and we rely on the motion of points that are on the skin, we must model the deformation of the hand skin when the hand moves, especially in areas close to joints.

The solution we propose solves for these three difficulties. To pass the first difficulty, the registration step is done using a robust version of the well-known Iterative Closest Point algorithm, EM-ICP [8], adapted to articulated and deformable object, followed by another adaptation of this method, EM-ICPS, where point-to-point and point-to-surface distances are taken into account. The second and third difficulties are solved by using an appropriate model: The hand is represented as a 27 DOF articulated model, with a Soft Objects model [14] to define the skin surface position, and Computer Animation *skinning* techniques to compute motion of points on the skin.

The rest of the paper is organized as follows:

- Section 2 presents the framework of tracking a rigid body of known shape (an ellipsoid in this case) from noisy 3D points trajectories using the EM-ICP and EM-ICPS algorithms.
- Section 3 presents the hand model we use, which is composed of an articulated model, a skin motion model, and a skin surface model. It also extends the tracking framework presented in Section 2 from rigid shapes to full articulated and deformable objects.
- Section 4 presents results of running the algorithm on a stereoscopic sequence of a moving hand, where a side view is used to evaluate the accuracy of this tracking method.

2 Tracking a Rigid Ellipsoid

The input data to our rigid body tracking method is composed of 3D points obtained from a stereoscopic image sequence, by running a points-of-interest detector in the left image, matching the detected points with the right image using a correlation criterion, and then either tracking those matched points over time [17], or repeating this process on each stereo pair. The classical image-based tracking procedure can be adapted to make use of the epipolar geometry as a constraint.

The object tracking problem can be expressed as follows: knowing the 3D object position at time t and optionally its kinematic screw (i.e. rotational and translational velocities), and having 3D points tracked between t and $t + 1$, find the object position at $t + 1$. If the point tracks are noise-free and do not contain outliers, there exists a direct solution to this problem, and in the presence of outliers and noisy data the ICP algorithm [23] and its derivatives [8,15] may solve this problem. In our case, we also have a surface model which can be used to get a better estimation of the motion. We call the resulting algorithm EM-ICPS (Expectation-Maximization Iterative Closest Point and Surface).

2.1 Description of the Ellipsoid

We first present this tracking framework on a rigid ellipsoid. We will later use ellipsoids as basic construction elements to build our hand model. We made this choice because we can easily derive a closed-form pseudo-distance from any point in space to the surface of this object. An ellipsoid with its axes aligned with the standard frame axes, and with axis lengths of respectively a , b , and c , can be described, using the 4×4 diagonal quadratic form $\mathcal{Q} = \text{diag}(\frac{1}{a^2}, \frac{1}{b^2}, \frac{1}{c^2}, -1)$, by the implicit equation of its surface $\mathbf{X}^T \mathcal{Q} \mathbf{X} = 0$, where $\mathbf{X} = (x, y, z, 1)$ is a point in space represented by its homogeneous coordinates. Similarly, it can be easily shown that the transform of this ellipsoid by a rotation \mathcal{R} and a translation \mathbf{t} can be described by the implicit equation:

$$q(\mathbf{X}) = \mathbf{X}^T \mathcal{Q}_{\mathcal{T}} \mathbf{X} = 0, \quad \text{where} \quad \mathcal{T} = \begin{pmatrix} \mathcal{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathcal{Q}_{\mathcal{T}} = \mathcal{T}^{-T} \mathcal{Q} \mathcal{T}^{-1}. \quad (1)$$

If two axes of the ellipsoid have equal length (by convention, $a = b$), it degenerates to a spheroid. Let \mathbf{C} be its center, let \mathbf{w} be a unit vector on its symmetry axis, and let \mathbf{u} and \mathbf{v} be two vectors forming an orthonormal frame $(\mathbf{u}, \mathbf{v}, \mathbf{w})$, the implicit surface equation becomes:

$$q(\mathbf{X}) = \frac{1}{a^2} \left(|\mathbf{CX}|^2 - (\mathbf{w} \cdot \mathbf{CX})^2 \left(1 - \frac{a^2}{c^2} \right) \right) - 1, \quad (2)$$

Computing the Euclidean distance from a point in space to the ellipsoid requires solving a sixth degree polynomial (fourth degree for a spheroid), thus we prefer using an approximation of the Euclidean distance which is the distance from the 3D point \mathbf{X} to the intersection \mathbf{R} of the line segment \mathbf{CX} and the

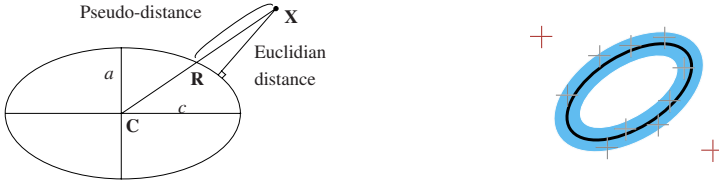


Fig. 1. Left: The pseudo-distance $d'(\mathbf{X})$ between a point \mathbf{X} and an ellipsoid is the distance $|\mathbf{XR}|$ to the intersection \mathbf{R} of the line segment \mathbf{CX} and the ellipsoid. Right: 3D points whose distance to the ellipsoid is below ϵ are used for tracking.

ellipsoid (Fig. 1). This pseudo distance is reasonably close to the Euclidean distance, except if the ratio between the smallest and the biggest axis length is very small. Since $|\mathbf{CX}| = |\mathbf{CR}|\sqrt{q(\mathbf{X}) + 1}$, the pseudo-distance $d'(\mathbf{X}) = |\mathbf{CX}| - |\mathbf{CR}|$ becomes, in the case of a spheroid:

$$d'(\mathbf{X}) = |\mathbf{CX}| - c \left(\sqrt{1 + \frac{(\mathbf{w} \cdot \mathbf{CX})^2}{|\mathbf{CX}|^2} \left(\frac{c^2}{a^2} - 1 \right)} \right)^{-1}. \quad (3)$$

2.2 Point-to-Point Tracking

The first method used to estimate the solid displacement over time consists in using only the motion of the 3D points. Let $\{\mathbf{X}_i\}_{i \in [1, N_t]}$ and $\{\mathbf{Y}_j\}_{j \in [1, N_{t+1}]}$ be the set of 3D points, respectively at times t and $t + 1$. This method must be able to take into account the fact that some points might be tracked incorrectly, either because of the image-based tracking process or because the tracked points do not belong to the same object. This means that some \mathbf{X}_i may not have a match among the \mathbf{Y}_j , and vice-versa.

The classical Iterative Closest Point algorithm [23] solves the problem of estimating the displacement $\mathcal{T}_{t,t+1}$ by iterating over the following 3-step procedure, starting with an estimate of $\mathcal{T}_{t,t+1}$ (which can be the identity if no prediction on the position is available):

1. For each \mathbf{X}_i , find its correspondent as the closest point to $\mathcal{T}_{t,t+1}\mathbf{X}_i$ in $\{\mathbf{Y}_j\}$.
2. Given the correspondences $(i, c(i))$, compute $\mathcal{T}_{t,t+1}$ which minimizes the objective function $E = \sum_{i \in [1, N_t]} |\mathcal{T}_{t,t+1}\mathbf{X}_i - \mathbf{Y}_{c(i)}|^2$ (there exists a closed-form solution in the case of a rigid displacement).
3. If increments in displacement parameters exceed a threshold, go to step 1.

The ICP algorithm may take into account points which have no correspondent among the $\{\mathbf{Y}_j\}$ by simply using a maximum distance between points above which they are rejected, but this may result in instabilities in the convergence process. A better formulation was proposed both by Granger and Pennec as

EM-ICP [8], and by Rangarajan *et al.* as SoftAssign [15]. It consists in taking into account the probability α_{ij} of each match in the objective function:

$$E_p = \sum_{(i,j) \in [1, N_t] \times [1, N_{t+1}]} \alpha_{ij} d_{ij}^2, \quad \text{with } d_{ij} = |\mathcal{T}_{t,t+1} \mathbf{X}_i - \mathbf{Y}_j|. \quad (4)$$

The probability α_{ij} is either 0 or 1 for ICP, and is between 0 and 1 for EM-ICP and SoftAssign (we detail the computation of α_{ij} later on). EM-ICP and SoftAssign¹ replace steps 1 and 2 of the above procedure by:

1. For $(i, j) \in [1, N_t] \times [1, N_{t+1}]$, compute the probability α_{ij} that \mathbf{X}_i matches \mathbf{Y}_j , using the distance $d_{ij} = |\mathcal{T}_{t,t+1} \mathbf{X}_i - \mathbf{Y}_j|$.
2. Compute $\mathcal{T}_{t,t+1}$ which minimizes the objective function of equation (4).

We now detail our implementation of EM-ICP, by answering the following questions: Which points should be taken into account? How do we compute the α_{ij} ? Once we have the α_{ij} , how do we re-estimate $\mathcal{T}_{t,t+1}$?

Which points should be taken into account? Since there may be multiple objects to track in the scene, we only take into account points at time t that are within some distance of the previous object position (Figure 1). This distance should be more than the modelling error, i.e. the maximum distance between the real object and the object model. Let σ_{model} be the standard deviation of this modelling error, we select points that are within a distance of $2\sigma_{\text{model}}$ to the model (a few millimeters in the case of the hand model). For simplicity's sake, we use the pseudo-distance $d'(\mathbf{X})$ instead of the Euclidean distance.

How do we compute the α_{ij} ? Under a Gaussian noise hypothesis on the 3D points position, Granger and Pennec [8] showed that the probability of a match between two points has the form: $\alpha_{ij} = \frac{1}{C_i} \exp(-d_{ij}^2/\sigma_p^2)$, where σ_p is a characteristic distance which decreases over ICP iterations (p stands for “points”), and C_i is computed so that the sum of each row of the correspondence matrix $[\alpha_{ij}]$ is 1. In our case, we add to the set of points at $t+1$ a virtual point, which is at distance d_0 of all points. This virtual point² is here to allow the outliers in $\{\mathbf{X}_i\}$ to be matched to no point within $\{\mathbf{Y}_j\}$. This leads to the following expression for α_{ij} :

$$\alpha_{ij} = \frac{1}{C_i} e^{-\frac{d_{ij}^2}{\sigma_p^2}}, \quad \text{with } C_i = e^{-\frac{d_0^2}{\sigma_p^2}} + \sum_{k=1}^{N_{t+1}} e^{-\frac{d_{i,k}^2}{\sigma_p^2}}. \quad (5)$$

¹ In fact, EM-ICP and SoftAssign are very close in their formulation, and the main difference is that SoftAssign tries to enforce a one-to-one correspondence between both sets by normalizing both rows and columns of the correspondence matrix $[\alpha_{ij}]$, whereas in EM-ICP the $\{\mathbf{Y}_j\}$ can be more dense than the $\{\mathbf{X}_i\}$, and only the rows of $[\alpha_{ij}]$ are normalized using the constraint $\sum_{i \in [1, N_t]} \alpha_{ij} = 1$. Both methods can take into account outliers, but our experiments showed that SoftAssign can easily converge to the trivial solution where all point are outliers, so that the we chose a solution closer to EM-ICP.

² Granger and Pennec did not need this virtual point in their case because the $\{\mathbf{Y}_j\}$ was an oversampled and augmented version of the $\{\mathbf{X}_i\}$, but Rangarajan *et al.* used a similar concept.

d_0 is the distance *after complete registration* between $\mathcal{T}_{t,t+1}\mathbf{X}_i$ and the $\{\mathbf{Y}_j\}$ above which this point is considered to have no match. Typically, it should be 2 to 3 times the standard deviation σ_{recons} of the noise on 3D point reconstruction (a few millimeters in our case). For the first iteration σ_p should be close to the typical 3D motion estimation error $\sigma_p^0 = \sigma_{\text{motion}}$ (1-2cm in our case), then should decrease in a few iterations to a value σ_p^∞ close to σ_{recons} , and should never go below this. We chose a geometric decrease for σ_p to get to $\sigma_p^\infty = \sigma_{\text{recons}}$ within 5 ICP iterations.

How do we re-estimate $\mathcal{T}_{t,t+1}$? Equation (4) can be rewritten as:

$$E_p = \sum_{i=1}^{N_t} \lambda_i^2 |\mathcal{T}_{t,t+1}\mathbf{X}_i - \mathbf{Z}_i|^2, \text{ with } \lambda_i = \sum_{j=1}^{N_{t+1}} \sqrt{\alpha_{ij}} \text{ and } \mathbf{Z}_i = \frac{1}{\lambda_i} \sum_{j=1}^{N_{t+1}} \sqrt{\alpha_{ij}} \mathbf{Y}_j. \quad (6)$$

With this reformulation, our objective function looks the same as in traditional ICP, and there exists a well-known closed-form solution for $\mathcal{T}_{t,t+1}$: the translation part is first computed as the barycenter of the $\{\mathbf{Z}_i\}$ with weights λ_i , and the remaining rotation part is solved using quaternions [23]. If an \mathbf{X}_i is an outlier, the corresponding λ_i will progressively tend to zero over iterations, and its effect will become negligible in E_p .

Caveats of point-to-point tracking: The main problem with points-based tracking is that either the image-based tracking or the ICP algorithm may drift over time because of accumulated errors, and nothing ensures that the points remain on the surface of the tracked object. For this reason, we had to incorporate the knowledge of the object surface into the ICP-based registration process.

2.3 Model-Based Tracking

In order to get a more accurate motion estimation, we introduce a tracking method where the points $\{\mathbf{Y}_j\}$ reconstructed at $t + 1$ are registered with the object model transformed by the estimated displacement $\mathcal{T}_{t,t+1}$. Previous approaches, when applied to model-based ICP registration, usually do this by using lots of sample points on the model surface, generating a huge computation overhead, especially in the search for nearest neighbors. We can use instead directly a distance function between any point in space and a single model. We choosed the pseudo-distance $d'(\mathbf{Y}_j)$ between points and the ellipsoid model.

In order to eliminate outliers (points that were tracked incorrectly or do not belong to the model), we use the same scheme as for point-to-point registration: the weight given to each point in the objective function E_s depends on a characteristic distance σ_s (where s stands for “surface”) which decreases over iterations from the motion estimation error $\sigma_s^0 = \sigma_{\text{motion}}$ to a lower limit σ_s^∞ :

$$E_s = \sum_j \beta_j d'(\mathbf{Y}_j)^2, \text{ with } \beta_j = e^{-\frac{d'(\mathbf{Y}_j)^2}{\sigma_s^2}}. \quad (7)$$

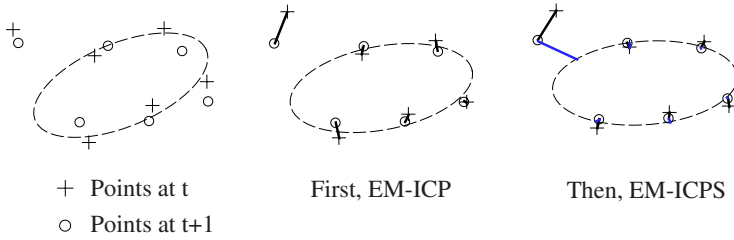


Fig. 2. The hybrid tracking method: A first registration is done using only point-to-point distances, and it is refined using both point-to-point and point-to-surface distances.

The lower limit of σ_s should take into account both the modeling error σ_{model} , i.e. the standard deviation of the distance between the model and the real object after registration, and the 3D points reconstruction error σ_{recons} . Since these are independent variables, it should be $\sigma_s^\infty = \sqrt{\sigma_{\text{model}}^2 + \sigma_{\text{recons}}^2}$.

The iterative procedure for model-based tracking is:

1. Compute the weights β_j as in equation (7).
2. Compute $\mathcal{T}_{t,t+1}$ which minimizes the objective function E_s .
3. If increments in displacement parameters exceed a threshold, go to step 1.

In general, there exists no closed-form solution to solve for step 2, so that a nonlinear least-squares method such as Levenberg-Marquardt has to be used. Of course, one could argue that the closed-form solution for step 2 of ICP will be much faster, but since step 2 uses the distance to the surface (and not a point-to-point distance), only 2 or 3 iterations of the whole procedure will be necessary. Comparatively, using standard ICP with a sampled model will require much more iterations.

Caveats of model-based tracking: The main problem with model-based tracking is that the model may “slide” over the points set, especially if the model has some kind of symmetry or local symmetry. This is the case with the spheroid, where the model can rotate freely around its symmetry axis.

2.4 Mixing Point-to-Point and Model-Based Tracking

In order to take advantage of both the EM-ICP and the model-based tracking, we propose to mix both methods (Figure 2). Since model-based tracking is computationally expensive, we first compute an estimation of $\mathcal{T}_{t,t+1}$ using EM-ICP. Then, we apply an hybrid procedure, which we call EM-ICPS (EM-Iterative Closest Point and Surface) to register the tracked points with the model:

1. For $(i, j) \in [1, N_t] \times [1, N_{t+1}]$, compute the probabilities α_{ij} as in equation (5), using $\sigma_p = \sigma_p^\infty$, and compute the weights β_j as in equation (7), using $\sigma_s = \sigma_s^\infty$.

2. Compute $\mathcal{T}_{t,t+1}$ which minimizes the objective function $E_{ps} = \omega_p E_p + \omega_s E_s$.
3. If increments in displacement parameters exceed a threshold, go to step 1.

The weights in E_{ps} are computed at first step to balance the contribution of each term: $\omega_p = 1/E_p^0$ and $\omega_s = 1/E_s^0$.

Bootstrapping the tracking method is always a difficult problem: How do we initialize the model parameters at time $t = 0$? In our case, we only need approximate model parameters, and we can simply adjust the model to the data at time 0 using our model-based registration procedure (section (2.3)).

3 Hand Model: Structure, Surface, Skin, and Tracking

This tracking procedure can be adapted to any rigid or deformable model. We need to be able to evaluate, for any position of the object, the distance between its surface and any point in 3D space (for the model-based tracking part) and a way to compute the movement of points on the surface of the object from the variation of the parameters of the model.

For this purpose, we developed an articulated hand model, where the skin surface is described as an implicit surface, based on Soft Objects [22] (Fig. 3), and the skin motion is described using skinning techniques as used in computer animation. We limited our work to the hand itself, but any part of the body could be modeled this way, and the model could be used using the same tracking technique, since no hand-specific assumption is done.

3.1 Structure

The basic structure of the hand is an articulated model with 27 degrees of freedom (Fig. 3), which is widely used in vision-based hand motion captures [19,5]. The joints in this model are either pivot or Cardan joints. Using the corresponding kinematic chain, one can compute the position in space \mathcal{T}^k of each phalanx, i.e. its rotation \mathcal{R}^k and its translation \mathbf{t}^k from the position \mathcal{T}^0 of the palm and the various rotation angles (21 in total).

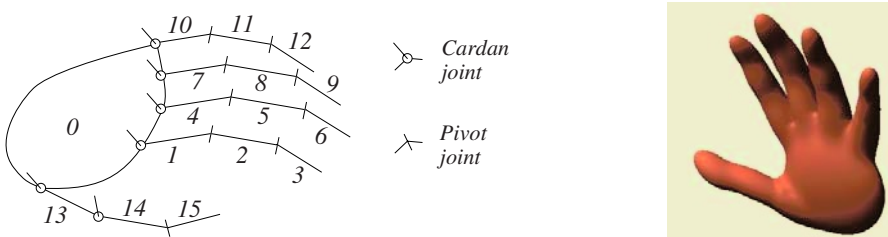


Fig. 3. Left: The 27 DOF model used for the hand structure: 16 parts are articulated using 6 Cardan joints and 9 pivot joints. Right: The final hand model.

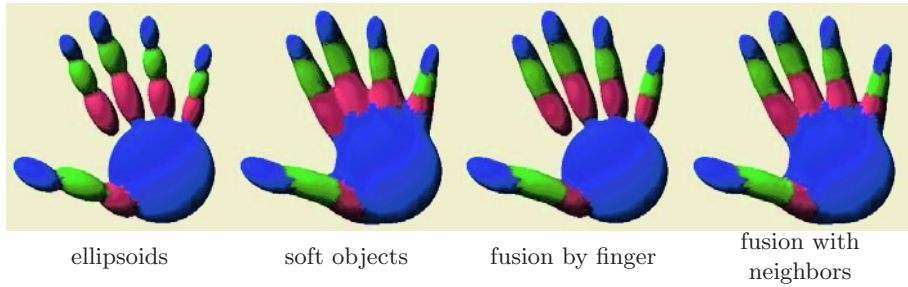


Fig. 4. Different ways to model the surface: using spheroids only, fusing all the spheroids using soft objects, fusing spheroids belonging to the same finger, and fusing each spheroid with its neighbors in the structure.

3.2 Surface Model Using Soft Objects

The surface model is used in the model-based registration part of our method. To model the hand, we choosed an implicit surface model, rather than mesh-based ou parametric models, since it’s possible, when the implicit function is carefully chosen, to easily compute the distance of any point in space to the implicit surface by taking the implicit function at this point.

Implicit functions can be described by models initially developed for Computer Graphics applications such as Meta-balls, Blinn Blobs [2], or Soft Objects [22]. These modeling tools were used recently for vision-based human motion tracking [14]. Very complex shapes can be created with few descriptors (as little as 230 meta-balls to obtain a realistic human body model [14]). We used one spheroid-based meta-ball per part (palm or phalanx), which makes a total of 16 meta-balls for the hand. Modeling an object using meta-balls is as easy as drawing ellipses on images, and that is how our hand model was constructed from a sample pair of images.

In these models, each object part is described by a carefully-chosen implicit function $f_k(\mathbf{X}) = 1$, and the whole object surface is obtained by fusing those parts together, considering the global implicit equation $f(\mathbf{X}) = \sum f_k(\mathbf{X}) = 1$. Most meta-balls models use the quadratic equation of the ellipsoid as the implicit function: $f_k(\mathbf{X}) = e^{-q_k(\mathbf{X})/\sigma^2}$. However, we found this function highly anisotropic, and the big ellipsoids get a larger influence (in terms of distance), which results in smaller ones being “eaten”. We preferred using our pseudo-distance as our base function:

$$f(\mathbf{X}) = \sum_{k=0}^{15} f_k(\mathbf{X}) = 1, \quad \text{with} \quad f_k(\mathbf{X}) = e^{-\frac{d'_k(\mathbf{X})}{\nu_k}}, \quad (8)$$

where ν_k is the distance of influence³ of each ellipsoid. Using the full sum can cause the fingers to stuck together when they are close (Fig. 4), so we reduced

³ ν_k is expressed in real world units, whereas σ in the classical meta-balls function is unit-less, resulting in the aforementioned unexpected behaviors.

the sum in eq. (8) to the closest ellipsoid and its neighbors in the hand structure described in Fig. 3. The resulting discontinuities of $f(\mathbf{X})$ are unnoticeable in the final model.

A pseudo-distance function can also be derived from the implicit function. In this case, a near-Euclidean distance is $d''(\mathbf{X}) = \ln(f(\mathbf{X}))$, which is valid even for points that are far away from the implicit surface, but a good approximation for points that are near the surface is simply⁴ $d''(\mathbf{X}) = f(\mathbf{X}) - 1$.

3.3 Skin Model

In order to use EM-ICP, we need to be able to deduct from the changes of the parameters describing our model the movement of any point at the surface of the object. This problem is addressed by skinning techniques used in computer animation [3]. Simply assessing that each point moves rigidly with the closest part of the hand structure would be insufficient nears joints where skin expands, retracts, or folds when the parameters change. So to compute the movement of a point on the surface, we used a classical linear combination of the motion of the closest part p , and this of the *second* closest part p' :

$$\mathcal{T}(\mathbf{X}) = \frac{f_p(\mathbf{X})^\alpha \mathcal{T}^p \mathbf{X} + f_{p'}(\mathbf{X})^\alpha \mathcal{T}^{p'} \mathbf{X}}{f_p(\mathbf{X})^\alpha + f_{p'}(\mathbf{X})^\alpha}. \quad (9)$$

The parameter α controls the behaviour of points that are near the joints of the model (Fig. 5): $\alpha = 2$ seemed to give the most realistic behaviour, both on the exterior (the skin is expanded) and on the interior (the skin folds) of a joint⁵.

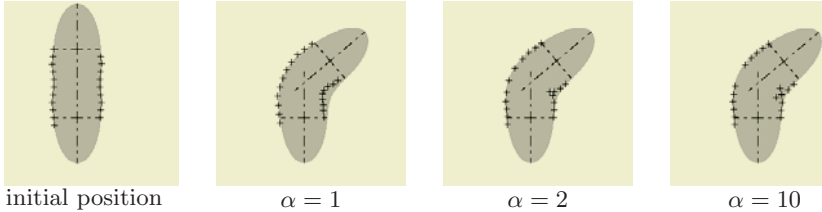


Fig. 5. The motion of points on the surface of the object depends on the choice of α , and does not exactly follow the implicit surface (in grey).

3.4 Tracking the Full Hand Model

There exist non-rigid variations of the ICP algorithm and its derivatives [7,4], but none exists for articulated objects, whereas our method works for an articulated

⁴ Simply because $f(\mathbf{X})$ is close to 1 near the surface, and $\ln(x+1) = x + O(x^2)$.

⁵ The fact that the skin motion does not exactly follow the implicit surface is not an issue, since both models are involved in different terms of the objective function E_{ps} .

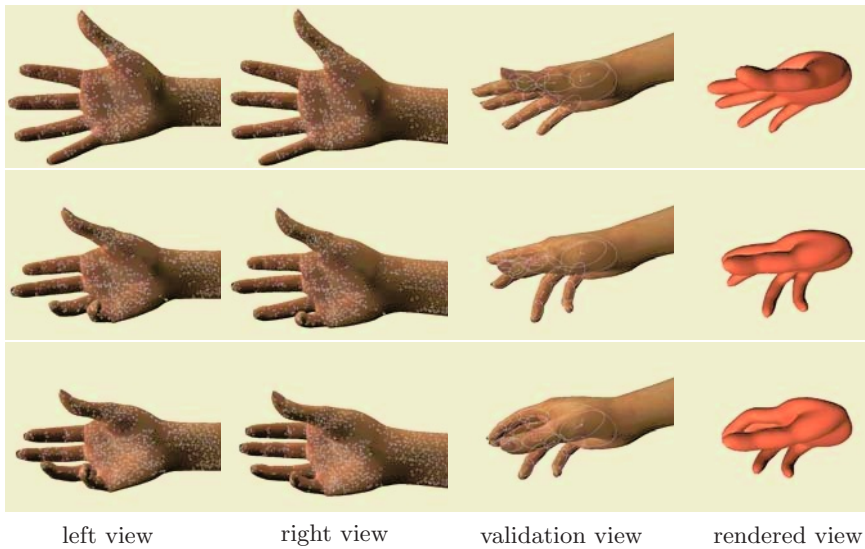


Fig. 6. Tracking results at frame 10, 30 and 50: left and right view with 3D points superimposed (views used in the tracking process), side view with each phalanx superimposed as an ellipse, and tracked model rendered using the camera parameters of the side view with marching cubes

and deformable object. The objective function to be optimized for the full hand model is (equations (4), (6) and (7)):

$$E_{ps} = E_p + E_s, \text{ with } E_s = \sum_j \beta_j (f(\mathbf{Y}_j) - 1)^2, \text{ and } \beta_j = e^{-\frac{(f(\mathbf{Y}_j) - 1)^2}{\sigma_s^2}}. \quad (10)$$

We look for the 27 parameters of the hand that minimize E_{ps} . E_p takes into account the skin model through the displacement field $\mathcal{T}_{t,t+1}$, while E_s uses the pseudo-distance to the implicit surface $d''(\mathbf{X}) = f(\mathbf{X}) - 1$. The optimization is done iteratively, using the procedure previously described in section 2.4, optimizing point-to-point distance first, and then point-to-point and point-to-surface distances.

4 Results and Conclusion

For our experiments, a stereoscopic sequence of 120 images was taken, where the palm is facing the cameras. About 500 points of interest were extracted and matched between the left and the right view, and the resulting 3D points served as input for our hand tracking. The hand structure was modeled interactively, by drawing the spheroids and the position of joints on the first image pair. A third view (side view) was used for the validation of tracking results. Figure 6 shows three time frames of the sequence, with the results superimposed on the

side view and the full model rendered with marching cubes in the same camera as the side view. All the fingers are tracked correctly, even those that are folded during the sequence, but we can see in the rendered view that the palm seems much thicker than it really is. This comes from the fact that we attached a *single* spheroid to each part, and the curvature of the palm cannot be modeled properly that way. In future versions, we will be able to attach several spheroids to each part of the model to fix this problem.

4.1 Conclusion

We presented a method for hand tracking from 3D point trajectories, which uses a full hand model, containing an articulated structure, a smooth surface model, and a realistic skin motion model. This method uses both the motion of points and the surface model in the registration process, and generalizes previous extensions of the Iterative Closest Point algorithm to articulated and deformable objects. It was applied to hand tracking, but should be general enough to be applied to other articulated and deformable objects (such as the whole human body). The results are quite satisfactory, and the method is being improved to handle more general hand motion (e.g. where the hand is flipped or the fist is clenched during the sequence). The tracking method in itself will probably be kept as-is, but improvements are being made on the input data (tracked points), and we will consider non-isotropic error for 3D reconstruction in the computation of the distance between two points. We are also working on automatic modeling, in order to get rid of the interactive initialization.

References

1. V. Athitsos and S. Sclaroff. Estimating 3D hand pose from a cluttered image. In *Proc. CVPR* [10].
2. J. Blinn. A generalization of algebraic surface drawing. *ACM Trans. on Graphics*, 1(3), 1982.
3. J. Bloomenthal. Medial based vertex deformation. In *Symposium on Computer Animation*, ACM SIGGRAPH, 2000.
4. H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *CVIU*, 89(2-3):114–141, Feb. 2003.
5. Q. Delamarre and O. Faugeras. Finding pose of hand in video images: a stereo-based approach. In *Third International Conference on Automatic Face and Gesture Recognition*, pages 585–590, Apr. 1998.
6. Q. Delamarre and O. Faugeras. 3D articulated models and multiview tracking with physical forces. *CVIU*, 81(3):328–357, Mar. 2001.
7. J. Feldmar and N. Ayache. Rigid, affine and locally addinne registration of free-form surfaces. *IJCV*, 18(2):99–119, May 1996.
8. S. Granger and X. Pennec. Multi-scale EM-ICP: A fast and robust approach for surface registration. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *ECCV*, volume 2353 of *LNCS*, pages 418–432, Copenhagen, Denmark, 2002. Springer-Verlag.

9. T. Heap and D. Hogg. Towards 3D hand tracking using a deformable model. In *Proc. Conf. on Automatic Face and Gesture Recognition*, pages 140–145, 1995.
10. IEEE Comp.Soc. Madison, Wisconsin, June 2003.
11. S. Lu, D. Metaxas, D. Samaras, and J. Oliensis. Using multiple cues for hand tracking and model refinement. In *Proc. CVPR* [10], pages 443–450.
12. T. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *CVIU*, 81(3):231–268, 2001.
13. K. Nirei, H. Saito, M. Mochimaru, and S. Ozawa. Human hand tracking from binocular image sequences. In *22th Int'l Conf. on Industrial Electronics, Control, and Instrumentation*, pages 297–302, Taipei, Aug. 1996.
14. R. Plänkers and P. Fua. Articulated soft objects for multi-view shape and motion capture. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(10), 2003.
15. A. Rangarajan, H. Chui, and F. L. Bookstein. The softassign procrustes matching algorithm. In *IPMI*, pages 29–42, 1997.
16. J. M. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Proc. 5th International Conference on Computer Vision*, pages 612–617, Boston, MA, June 1995. IEEE Comp.Soc. Press.
17. J. Shi and C. Tomasi. Good features to track. In *Proc. CVPR*, pages 593–600, Seattle, WA, June 1994. IEEE.
18. N. Shimada, K. Kimura, and Y. Shirai. Real-time 3-D hand posture estimation based on 2-D appearance retrieval using monocular camera. In *Proc. 2nd Int'l Workshop Recognition, Analysis, and Tracking of Faces and Gestures in Realtime Systems (RATFG-RTS)*, pages 23–30, Vancouver, Canada, July 2001.
19. B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model based 3D tracking of an articulated hand. In *Proc. CVPR*, pages 310–315. IEEE Comp.Soc., 2001.
20. A. Utsumi and J. Ohya. Multiple-hand-gesture tracking using multiple cameras. In *Proc. CVPR*, pages 1473–1478, Fort Collins, Colorado, June 1999. IEEE Comp.Soc.
21. Y. Wu, J. Y. Lin, and T. S. Huang. Capturing natural hand articulation. In *Proc. 8th International Conference on Computer Vision*, pages 426–432, Vancouver, Canada, 2001. IEEE Comp.Soc., IEEE Comp.Soc. Press.
22. B. Wyvill, C. McPheeters, and G. Wyvill. Data structure for soft objects. *The Visual Computer*, 2(4):227–234, 1986.
23. Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *IJCV*, 13(2):119–152, 1994.

A Robust Probabilistic Estimation Framework for Parametric Image Models^{*}

Maneesh Singh, Himanshu Arora, and Narendra Ahuja

University of Illinois at Urbana-Champaign, Urbana 61801, USA,
{msingh,harora1,n-ahuja}@uiuc.edu,
<http://vision.ai.uiuc.edu/~msingh>

Abstract. Models of spatial variation in images are central to a large number of low-level computer vision problems including segmentation, registration, and 3D structure detection. Often, images are represented using parametric models to characterize (noise-free) image variation, and, additive noise. However, the noise model may be unknown and parametric models may only be valid on individual segments of the image. Consequently, we model noise using a nonparametric kernel density estimation framework and use a locally or globally linear parametric model to represent the noise-free image pattern. This results in a novel, robust, redescending, M- parameter estimator for the above image model which we call the Kernel Maximum Likelihood estimator (KML). We also provide a provably convergent, iterative algorithm for the resultant optimization problem. The estimation framework is empirically validated on synthetic data and applied to the task of range image segmentation.

1 Introduction

Models of spatial variation in visual images are central to a large number of low level computer vision problems. For example, segmentation requires statistical models of variation within individual segments. Registration requires such models for finding correspondence across images. Image smoothing depends on distinction between bona fide image variation versus noise. Achieving super-resolution critically depends on the nature of photometric variation underlying an image of a given resolution. A general image model consists of two components - one that captures noise-free image variation, and the other that models additive noise. Robust solutions to low level vision problems require specification of these components. Local or piecewise variation can often be modelled in a simple parametric framework. However, recognizing that a model is applicable only in a specific neighborhood of a data point is a challenge. Added to this is the complex and data dependent nature of noise. Robust estimation of model parameters must address both of these challenges. As concrete examples of these challenges, (a) consider the problem of estimating the number of planar patches

^{*} The support of the Office of Naval Research under grant N00014-03-1-0107 is gratefully acknowledged.

in a range image. One cannot compute a least squares planar fit for the entire range image data as there might be several planes¹. (b) Further, range data may be corrupted with unknown additive noise. In this paper we use linear parametric models to represent local image variation and develop a robust framework for parameter estimation.

Much work has been done on robust parameter estimation in statistics and more recently in vision and it is difficult to refer to all of it here. We point the reader to some important papers in this area. The seminal paper by Huber [1] defined M-estimators (maximum-likelihood like estimators) and studied their asymptotic properties for linear models. This analysis is carried forward by Yohai and Maronna [2], and Koenker and Portnoy [3], among others. Recently, Chu et al. [4] introduced a redescending M-estimator for robust smoothing that preserves image edges (though the estimator is not consistent). Hillebrand and Müller [5] modified this estimator and proved its consistency. In computer vision, M-estimation has been extensively used for video coding [6], optical flow estimation [7], pose estimation [8], extraction of geometric primitives from images [9] and segmentation [10]. For a review on robust parameter estimation in computer vision, refer to Stewart [11]. More recently, the problem of robust estimation of local image structure have been extensively studied by [12,13,14], however they implicitly assume a locally constant image model while our model is more general.

Our main contributions are: (A) We show that a redescending M-estimator for linear parametric models can be derived through Hyndman et al.'s [15] modification to kernel regression estimators that reduces its bias to zero. This provides a link between nonparametric density estimation and parametric regression. It is illustrative to compare our approach with Chen and Meer [16], who use a geometric analogy to derive the estimator whereas we use the maximum likelihood approach. (B) The solution to the M-estimator (under mild conditions) is a nonlinear program (NLP). We provide a fast, iterative algorithm for the NLP and prove its convergence properties. (C) We show the utility of our algorithm for the task of segmentation of piecewise-planar range images.

In Section 2, we define the problem of robust linear regression. In Section 3, we first present the likelihood framework for parameter estimation using zero bias-in-mean kernel density estimators and then we provide a solution to the estimation problem. In Section 4, we empirically evaluate the performance of the proposed estimator and compare it with the least squares and least trimmed squares estimators under various noise contaminations. Finally, in Section 5, we show an instance of a vision problem where the algorithm can be applied.

¹ One may point out the use of the Hough transform for this task. However, the Hough transform is a limiting case of the robust parameter estimation model that we develop here.

2 Linear Regression

In [17], we presented a kernel density estimation framework for the image model,

$$Y(\mathbf{t}) = I(\mathbf{t}) + \epsilon(\mathbf{t}) \quad (1)$$

where $Y(\cdot)$, the observed image, is a sum of the clean image signal $I : \mathcal{R}^d \rightarrow \mathcal{R}$, and a noise process $\epsilon : \mathcal{R}^d \rightarrow \mathcal{R}$. In this paper, we use a linear parametric model for the image signal. More specifically, $I(\cdot)$ is either locally or globally linear in model parameters. We assume that the noise is unknown. This relaxation in assumptions about noise allows us to develop robust models for parameter estimation.

Definition 1 (Linear Parametric Signal Model) *The signal model in (1) is called linear in the model parameters, $\Theta \doteq [\theta_1, \theta_2, \dots, \theta_d]^T$, if the function $I(\mathbf{t})$ can be expressed as $I(\mathbf{t}) \doteq \mathbf{g}(\mathbf{t})^T \Theta = \sum_{i=1}^d \theta_i g_i(\mathbf{t})$ for some $\mathbf{g}(\mathbf{t}) \doteq [g_1(\mathbf{t}), \dots, g_d(\mathbf{t})]$.*

The problem of linear regression estimation is to estimate the model parameters $\Theta = [\theta_1, \dots, \theta_d]^T$ given a set of observations² $\{\mathbf{z}_i \doteq [Y_i; \mathbf{t}_i]\}_{i=1}^n$. One popular framework is to use least squares (LS) estimation where the estimate minimizes the mean square error between the sample values and those predicted by the parametric model.

Definition 2 (Least Squares Estimate) *Given a sample $\{\mathbf{z}_i\}_{i=1}^n$ and a fixed, convex set of weights $\{w_i\}_{i=1}^n$, the least weighted squares estimate $\hat{\Theta}_{LWS}$ for the linear parametric image model in Definition 1 is given by, $\hat{\Theta}_{LWS} = \min_{\Theta \in \mathcal{R}^d} \sum_{i=1}^n w_i \epsilon_i^2(\Theta)$ where $\epsilon_i(\Theta) \doteq Y_i - \mathbf{g}(\mathbf{t}_i)^T \Theta$. When the weights are all equal, then $\hat{\Theta}_{LWS}$ is the standard least squares estimate, denoted by $\hat{\Theta}_{LS}$.*

It is well known that the LS estimate is the maximal likelihood estimate if the errors, $\{\epsilon_i\}$, are zero-mean, independent and identically distributed with a normal distribution. In practice, however, the error distribution is not normal and often unknown. Since the error pdf is required for optimal parameter estimation and is often unknown, robust estimators have been used to discard some observations as outliers [11]. In this paper we formulate a maximum likelihood estimation approach based on the estimation of error pdf using kernel density estimators. Such an approach yields a robust redescending M-estimator. Our approach, however, has two advantages: (1) The formulation has a simple, convergent, iterative solution, and, (2) information from other sources (for example, priors on the parameters) can be factored in via a probabilistic framework.

² We use $[a; b]$ to denote a vertical concatenation of column vectors, i.e., $[a^T, b^T]^T$

3 Kernel Estimators for Parametric Regression

A conditional density estimator for image signals using probability kernels can be defined as follows (see [17] for properties and details),

Definition 3 (Kernel Conditional Density Estimator) *Let $\mathbf{z} = [Y; \mathbf{t}] = [Y, x_1, \dots, x_d]^T \in \mathcal{R}^d$ be a $(d+1)$ -tuple. Then, we write the kernel estimator for the conditional density of Y given the spatial location $\mathbf{t} = [x_1, \dots, x_d]^T$ as,*

$$\hat{f}_{Y|\mathbf{t}}(v|\mathbf{u}) = \frac{1}{n|H|\mathcal{D}} \sum_{i=1}^n K(H^{-1}[v - Y(\mathbf{t}_i); (\mathbf{u} - \mathbf{t}_i)]) \quad (2)$$

where H is a non-singular $(d+1) \times (d+1)$ bandwidth matrix and $K : \mathcal{R}^{d+1} \rightarrow \mathcal{R}$ is a kernel such that it is non-negative, has a unit area ($\int_{\mathcal{R}^{d+1}} K(\mathbf{z}) d\mathbf{z} = 1$), zero mean ($\int_{\mathcal{R}^{d+1}} \mathbf{z}K(\mathbf{z}) d\mathbf{z} = 0$), and, unit covariance ($\int_{\mathcal{R}^{d+1}} \mathbf{z}\mathbf{z}^T K(\mathbf{z}) d\mathbf{z} = I_{d+1}$).

For data defined on a regular grid, such as for images, $\mathcal{D} = \frac{1}{n|H|} \sum_{i=1}^n \int_{\mathcal{R}} K(H^{-1}(\mathbf{z}_i - \mathbf{z})) dY =: \frac{1}{n} \sum_{i=1}^n w_i(\mathbf{t})$ can be treated as a normalization constant which clearly does not depend upon the values $\{\mathbf{Y}_i\}_i$. $w_i(\mathbf{t})$ or simply w_i (when the spatial location \mathbf{t} is unambiguous) is given by $\frac{1}{|H|} \int_{\mathcal{R}} K(H^{-1}(\mathbf{z}_i - \mathbf{z})) dY$. We sometimes assume that the probability kernels are separable or rotationally symmetric or both. In the sequel, we require a specific kind of separability: we would like the spatial domain and the intensity domain to be separable. We shall also assume that the kernels are rotationally symmetric with a convex profile (please refer to Chapter 6, [18]).

3.1 Zero Bias-in-Mean Kernel Regression

Thus, we will assume kernel $K(\cdot)$ to be separable in the following sense: Let $\mathbf{z} \doteq [Y, \mathbf{t}^T]^T$ denote any data point such that the first coordinate is its intensity component and the rest are its spatial coordinates. Kernel K is separable in the subspaces represented by the intensity and spatial coordinates, denoted by \mathcal{D}_Y and $\mathcal{D}_{\mathbf{t}} \doteq \mathcal{R}^d \ominus \mathcal{D}_Y$. Assuming $H^{-1} = \begin{pmatrix} 1/h_Y & 0_{1 \times d} \\ 0_{d \times 1} & H_{\mathbf{t}}^{-1} \end{pmatrix}$, $K(H^{-1}\mathbf{z}) = K_Y(\frac{Y}{h_Y})K_{\mathbf{t}}(H_{\mathbf{t}}^{-1}\mathbf{t})$.

In (2), values $\{Y_i \doteq Y(\mathbf{t}_i)\}_i$ at locations $\{\mathbf{t}_i\}_i$ are used to *predict* the behavior of Y at location \mathbf{u} . For the kernel estimate, the expected value of Y conditioned on \mathbf{u} is given by, $E[Y|\mathbf{u}] = \sum_{i=1}^n w_i(\mathbf{u})$. Since conditional mean is independent of the regression model, we need to factor the model in Definition 1, into the conditional kernel density estimate in (2). The necessity for doing so is explained using Figure 1(a). At location $\mathbf{u} = 5$ shown on the horizontal axis, $E[Y|\mathbf{u}]$ will have a positive bias since most values, Y_i , in the neighborhood are larger than the *true* intensity value at $\mathbf{u} = 5$. The conditional mean will have this bias if the image is not locally linear. In general, the conditional density estimate will be biased wherever the image is not (locally) constant. This effect was noted

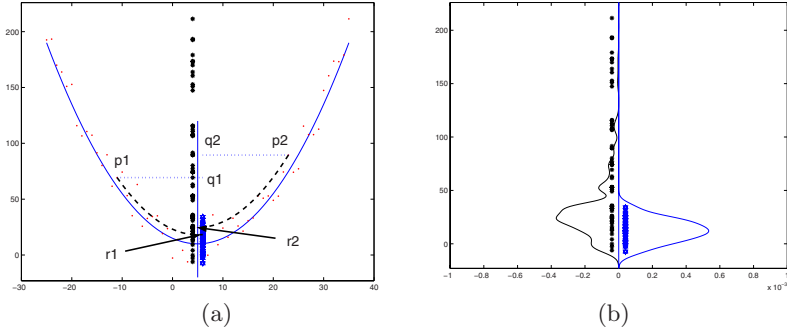


Fig. 1. This figure shows that kernel density estimation after factoring in the image model leads to better estimates. (a) Shown are the data points generated using the 1D image model $Y(t) = 10 + 0.2 * (t - 5)^2 + N(0, 10^2)$. To estimate the density $f(Y|t = 5)$ according to Definitions 3 and 4, data points $p1$ and $p2$ are moved along straight and curved lines respectively. (b) Shown are the two estimated pdfs, $\hat{f}_{Y|t}$ and $\tilde{f}_{Y|t}$ to the left and right respectively ($h_t = 10, h_Y = 5$).

by Hyndman et. al. [15]. To address this defect, they proposed the following modified kernel estimator which we call the zero bias-in-mean kernel (0-BIMK) conditional density estimator.

Definition 4 (0-BIMK Conditional Density Estimator) *The zero bias-in-mean kernel (0-BIMK) estimator for the conditional pdf is given by, $\hat{f}_{Y|t}(v|\mathbf{u}) \doteq \frac{1}{n|H|\mathcal{D}} \sum_{i=1}^n K(H^{-1}[v - \tilde{Y}(\mathbf{t}_i); (\mathbf{u} - \mathbf{t}_i)])$ such that, $\tilde{Y}_i = I(\mathbf{u}) + \epsilon_i - \bar{\epsilon}$, $\epsilon_i = Y_i - I(\mathbf{t}_i)$, $\bar{\epsilon} = \sum_{i=1}^n w_i \epsilon_i$.*

The conditional mean of Y using the above density estimator is, $\tilde{E}[Y|\mathbf{x}] = \sum_{i=1}^n w_i \tilde{Y}_i = I(\mathbf{x})$. Thus, the conditional mean of Y computed using the 0-BIMK estimator is guaranteed to be that predicted by the model since the conditional density at any domain point is computed by factoring in the assumed underlying model. In Figure 1(a), we show how the density estimate is constructed using points $p1$ and $p2$. While intensity values $q1$ and $q2$ (after spatial weighing) are used to find $\hat{f}_{Y|t}$ at $t = 5$, $r1$ and $r2$ are used to estimate $\tilde{f}_{Y|t}$. The two estimates are depicted in Figure 1(b) - among them, the latter estimate is clearly a *better* estimate of $10 + N(0, 10^2)$.

3.2 0-BIM Parametric Kernel Regression

We are interested in parametric regression models where the regression model, $I(t)$, has the form $I(\mathbf{t}) = \Theta^T \mathbf{g}(\mathbf{t})$. Our goal is to estimate the model parameters, Θ and thus, the regression function, $I(\mathbf{t})$. Let the parameter estimate be $\hat{\Theta}$. Then, the regression estimate is given by $\hat{I}(\mathbf{t}) = \hat{\Theta}^T \mathbf{g}(\mathbf{t})$. Using Definition 4, the kernel density estimator using the parametric regression model is defined below.

Definition 5 (0-BIMK Parametric Density Estimator) *The zero bias-in-mean kernel (0-BIMK) conditional density estimator, using the parametric regression model $I(\mathbf{t}) = \Theta^T \mathbf{g}(\mathbf{t})$ in Definition 1, is given by,*

$$\tilde{f}_{Y|\mathbf{t}}(v|\mathbf{u}, \Theta) \doteq \frac{1}{n|H|\mathcal{D}} \sum_{i=1}^n K(H^{-1}[v - \tilde{Y}(\mathbf{t}_i); (\mathbf{u} - \mathbf{t}_i)]) \quad (3)$$

with $\tilde{Y}_i \doteq g(\mathbf{u})^T \Theta + \epsilon_i(\Theta) - \bar{\epsilon}(\Theta)$, $\epsilon_i \doteq Y_i - g(\mathbf{t}_i)^T \Theta$ and $\bar{\epsilon}(\Theta) \doteq \sum_{j=1}^n w_j \epsilon_j(\Theta)$.

The minimum variance parameter estimator (KMV) for the above density estimate is not a particularly good choice. It is easy to see that the KMV estimate chooses that pdf for the noise process which minimizes the estimated noise variance. This is clearly undesirable if the error pdf is multi-modal. In such a scenario, kernel pdf estimation becomes useful. It provides us with the knowledge that the data is multimodal and assists in choosing the correct mode for the parameter estimate.³ This prompts the following variant of the ML parameter estimation procedure (one might define other such criteria).

Definition 6 Kernel Maximum Likelihood Parameter Estimator, $\tilde{\Theta}_{KML}$ or simply, $\tilde{\Theta}$, is defined to be that value of Θ which maximizes the likelihood that $Y(\mathbf{t}) = I(\mathbf{t})$ when the likelihood is estimated using the 0-BIMK estimator.

$$\tilde{\Theta}_{KML} \doteq \max_{\Theta \in \mathcal{R}^d} \tilde{f}_{Y|\mathbf{t}}(I(\mathbf{t})|\mathbf{t}, \Theta) \quad (4)$$

3.3 Estimation of $\tilde{\Theta}$

$\tilde{\Theta}$ is the solution of an unconstrained and differentiable nonlinear program (NLP) which requires global optimization techniques. We introduce now an iterative nonlinear maximization algorithm that is guaranteed to converge to a local maximum for any starting point. The main attractiveness of our algorithm is that it chooses the step size automatically. The convergence result holds if the probability kernel, $K(\cdot)$, has a convex profile (please refer to Chapter 6, [18]). The algorithm can be repeated for several starting points for global maximization (a standard technique for global optimization). We build the proof of convergence using the following two results.

Theorem 7 (Least Squares Fit) *Given a data sample $\{[Y_i, \mathbf{t}_i^T]\}_{i=1}^n$, the Least Squares fit of the model $Y \approx \Theta^T g(\mathbf{t})$, defined by $\hat{\Theta}_{LWS}$ in Definition 2 is a solution of the normal equations, $\sum_{i=1}^n w_i g(\mathbf{t}_i) g(\mathbf{t}_i)^T \hat{\Theta}_{LS} = \sum_{i=1}^n w_i g(\mathbf{t}_i) Y_i$.*

³ The cost of this generalization achieved by kernel pdf estimation, however, is that one needs to know the scale of the underlying pdf at which it is to be estimated from its samples.

Lemma 8 *Let there be n points, $\{\mathbf{t}_i\}_{i=1}^n$, in \mathcal{R}^d , m functions, $g_i : \mathcal{R}^d \rightarrow \mathcal{R}$, $i = 1, \dots, m$ and strictly positive convex weights $\{w_i\}_{i=1}^n$. Then the matrix $A \doteq \sum_{i=1}^n w_i(g(\mathbf{t}_i) - \overline{g(\mathbf{t})})(g(\mathbf{t}_i) - \overline{g(\mathbf{t})})^T$ is invertible if and only if there exist n' distinct points, $n \geq n' \geq m$, such that the set of functions, $\{\sqrt{w_i}(g(\mathbf{t}_i) - \overline{g(\mathbf{t})})\}_{i=1}^m$, is independent over them.*

Proof. Let Q_A be the quadratic form associated with A . For every $\mathbf{y} \in \mathcal{R}^d$, $Q_A(\mathbf{y}) \doteq \mathbf{y}^T A \mathbf{y} \geq 0$ since $Q_A(\mathbf{y}) = \sum_{i=1}^n w_i((g(\mathbf{t}_i) - \overline{g(\mathbf{t})}) \cdot \mathbf{y})^2 \geq 0$. Thus, A is positive semidefinite. Hence A is invertible iff it is positive definite. Further, A is positive definite iff the set of vectors $\{\sqrt{w_i}(g(\mathbf{t}_i) - \overline{g(\mathbf{t})})\}_i$ spans \mathcal{R}^d . Evidently, this is possible iff there exist n' distinct points, such that $n \geq n' \geq m$ and the set of functions $\{\sqrt{w_i}(g(\mathbf{t}_i) - \overline{g(\mathbf{t})})\}_{k=1}^m$ are independent over such a set. \square

Theorem 9 (Parametric Mean Shift using Weighted Least Squares)

Let there be n points, $\{\mathbf{t}_i\}_{i=1}^n$, in \mathcal{R}^d such that at least m points are distinct. Further, let there be m independent functions, $g_i : \mathcal{R}^d \rightarrow \mathcal{R}$, $i = 1, \dots, m$ as in Definition 6. Also, let K be such that $K(H^{-1}\mathbf{z}) = K_Y(\frac{Y}{h_Y})K_t(H_t^{-1}\mathbf{t})$. If K_Y has a convex and non-decreasing profile κ , then the sequence $\{\tilde{f}(j)\}_{j=1,2,\dots} \doteq \{\tilde{f}_{Y|\mathbf{t}}(I(\mathbf{t})|\mathbf{t}, \Theta_j)\}_{j=1,2,\dots}$, of probability values computed at $\mathbf{z} = (I(\mathbf{t}), \mathbf{t})$ with corresponding parameter estimates $\{\Theta_j\}_{j=1,2,\dots}$ defined⁴ by $\Theta_{j+1} = \text{argzero}_{\Theta} A_j(\Theta - \Theta_j) - b_j$ according to (7) below, is convergent. The sequence $\{\Theta_j\}_{j=1,2,\dots}$ converges to a local maximum of $\{\tilde{f}_{Y|\mathbf{t}}(I(\mathbf{t})|\mathbf{t}, \Theta)\}$.

Proof. From Definition (5),

$$\begin{aligned} \tilde{f}(\Theta_j) &\doteq \tilde{f}_{Y|\mathbf{t}}(I(\mathbf{t})|\mathbf{t}, \Theta_j) = \frac{1}{n|H|\mathcal{D}} \sum_{i=1}^n K(H^{-1}[\epsilon_i(\Theta_j) - \bar{\epsilon}(\Theta_j); (\mathbf{u} - \mathbf{t}_i)]) \\ &= \frac{1}{h_Y} \sum_{i=1}^n K_Y\left(\frac{\epsilon_i(\Theta_j) - \bar{\epsilon}(\Theta_j)}{h_Y}\right) w_i \end{aligned}$$

Using convexity of the profile κ , defining $w_{i,j} = \kappa'(-\frac{(\epsilon_i(\Theta_j) - \bar{\epsilon}(\Theta_j))^2}{h_Y^2}) w_i$ and denoting all the multiplicative constants by C , we get⁵,

$$\begin{aligned} \tilde{f}(\Theta_{j+1}) - \tilde{f}(\Theta_j) &\geq C \sum_{i=1}^n w_{i,j} \{(\epsilon_{i,j}^c)^2 - (\epsilon_{i,j}^c - (\Theta_{j+1} - \Theta_j)^T(g(\mathbf{t}_i) - \overline{g(\mathbf{t})}))^2\} \\ (\text{where, } \epsilon_{i,j}^c &\doteq (Y_i - \bar{Y}) - \Theta_j^T(g(\mathbf{t}_i) - \overline{g(\mathbf{t})}) \text{ and } \overline{g(t)} \doteq \sum_i w_i g(t)) \end{aligned} \quad (5)$$

The above equation shows that the RHS and consequently, the LHS is non-negative if solution to the following LWS problem exists.

$$\Theta_{j+1} - \Theta_j \doteq \underset{\Theta \in \mathcal{R}^d}{\text{argmin}} \sum_{i=1}^n w_{i,j} (\epsilon_{i,j}^c - \Theta^T(g(\mathbf{t}_i) - \overline{g(\mathbf{t})}))^2$$

⁴ $y = \text{argzero}_x g(x)$ denotes a value of y such that $g(y) = 0$

⁵ Since $\tilde{f}(\Theta_{j+1}) - \tilde{f}(\Theta_j) \geq C \sum_{i=1}^n w_{i,j} ((\epsilon_i(\Theta_j) - \bar{\epsilon}(\Theta_j))^2 - (\epsilon_i(\Theta_{j+1}) - \bar{\epsilon}(\Theta_{j+1}))^2) = C \sum_{i=1}^n w_{i,j} (((Y_i - \bar{Y}) - \Theta_j^T(g(\mathbf{t}_i) - \overline{g(\mathbf{t})}))^2 - ((Y_i - \bar{Y}) - \Theta_{j+1}^T(g(\mathbf{t}_i) - \overline{g(\mathbf{t})}))^2)$

From Theorem (7), $\Theta_{j+1} - \Theta_j$ must satisfy the following normal equations,

$$\begin{aligned} \sum_{i=1}^n w_{i,j} (g(\mathbf{t}_i) - \overline{g(\mathbf{t})}) (g(\mathbf{t}_i) - \overline{g(\mathbf{t})})^T (\Theta_{j+1} - \Theta_j) \\ = \sum_{i=1}^n w_{i,j} \epsilon_{i,j}^c (g(\mathbf{t}_i) - \overline{g(\mathbf{t})}) = \nabla_{\Theta} \tilde{f}(\Theta_j) \end{aligned} \quad (6)$$

which has the form $A_j(\Theta_{j+1} - \Theta_j) = b_j$ where

$$A_j \doteq \sum_{i=1}^n w_{i,j} (g(\mathbf{t}_i) - \overline{g(\mathbf{t})}) (g(\mathbf{t}_i) - \overline{g(\mathbf{t})})^T; b_j \doteq \sum_{i=1}^n w_{i,j} \epsilon_{i,j}^c (g(\mathbf{t}_i) - \overline{g(\mathbf{t})}) \quad (7)$$

Further, Lemma (8) guarantees the existence of a solution. Hence, RHS of (5)

$$\begin{aligned} &= C \sum_{i=1}^n w_{i,j} \{2\epsilon_{i,j}^c (\Theta_{j+1} - \Theta_j)^T (g(\mathbf{t}_i) - \overline{g(\mathbf{t})}) - \{(\Theta_{j+1} - \Theta_j)^T (g(\mathbf{t}_i) - \overline{g(\mathbf{t})})\}^2\} \\ &= C \sum_{i=1}^n w_{i,j} \{(\Theta_{j+1} - \Theta_j)^T (g(\mathbf{t}_i) - \overline{g(\mathbf{t})})\}^2 = (\nabla_{\Theta} \tilde{f}(j))^T (\Theta_{j+1} - \Theta_j) \geq 0 \end{aligned}$$

where the equalities follow from (6). The above equation shows that $S_f \doteq \{\tilde{f}(\Theta_j)\}_j$ is a nondecreasing sequence. Since the sequence is obviously bounded, it is convergent. RHS of the third equality above also shows that the solution to the normal equations always produces a step which has a positive projection along the gradient of the estimated pdf. By Lemma (8), $A_j, j = 1, 2, \dots$ are fully ranked for all j 's. RHS of the second equality above also implies that $\|\delta\Theta_j\| \rightarrow 0$ where $\delta\Theta_j \doteq \Theta_{j+1} - \Theta_j$.

Now, we show that the sequence $S_{\Theta} \doteq \{\Theta_j\}_{j=1,2,\dots}$ is convergent⁶. As the sequence is bounded, it has an isolated (since n is finite) limit point Θ^* . Since $\delta\Theta_j \rightarrow 0$, given small enough r, ϵ , $0 < r < r + \epsilon$ and open balls $B(\Theta^*, r)$, $B(\Theta^*, r + \epsilon)$, there exists an index J_1 such that for all $j > J_1$, $\|\delta\Theta_j\| < \epsilon$, Θ^* is the only limit point in $B(\Theta^*, r + \epsilon)$ and the set $U \doteq \{\Theta | \Theta \in B(\Theta^*, r + \epsilon) \cap B^c(\Theta^*, r), \Theta \in S\}$ is non-empty. U has finite items, let their maximum value be M_U . Further, as S_f is strictly increasing, there exists an index J_2 such that for all $j > J_2$, $\tilde{f}(\Theta_j) > M_U$. We define $J = \max(J_1, J_2)$. Then, for any $j > J$, $\Theta_j \in B(\Theta^*, r) \Rightarrow \Theta_{j+1} \in B(\Theta^*, r)$ since $B(\Theta^*, r) \cup B(\Theta_j, \delta\Theta_j) \subset B(\Theta^*, r + \epsilon)$ and $\Theta_{j+1} \notin U$. Further, $\delta\Theta_j \rightarrow 0 \Rightarrow b_j = \nabla_{\Theta} \tilde{f}(\Theta_j) \rightarrow 0$. Hence, Θ^* is a point of local maximum for $\tilde{f}(\cdot)$. \square

Thus, Theorem (4) guarantees a solution to the kernel maximum likelihood problem in the sense that given any starting point, we would converge to a local maximum of the parametric kernel likelihood function. However, at each iteration step, a weighted least squares problem needs to be solved that requires matrix

⁶ We gratefully acknowledge Dr. Dorin Comaniciu's assistance with the proof.

inversion. Further, it is required that in the limit, the matrix A_j stays positive definite. This condition need not always be satisfied. In the next theorem, we provide a way to bypass this condition.

Theorem 10 (Parametric Mean Shift using Gradient Descent)

Let there be n points, $\{\mathbf{t}_i\}_{i=1}^n$, in \mathcal{R}^d such that at least m points are distinct. Further, let there be m independent functions, $g_i : \mathcal{R}^d \rightarrow \mathcal{R}$, $i = 1, \dots, m$ in terms of which we want to find $\tilde{\Theta}_{KML}$ as given in Definition 6. Also, let K be such that $K(H^{-1}\mathbf{z}) = K_Y(\frac{Y}{h_Y})K_{\mathbf{t}}(H_{\mathbf{t}}^{-1}\mathbf{t})$. If K_Y has a convex and non-decreasing profile κ , then the sequence $\{\tilde{f}(j)\}_{j=1,2,\dots} \doteq \{\tilde{f}_{Y|\mathbf{t}}(I(\mathbf{t})|\mathbf{t}, \Theta_j)\}_{j=1,2,\dots}$ of probability values computed at $\mathbf{z} = (I(\mathbf{t}), \mathbf{t})$ with corresponding parameter estimates $\{\Theta_j\}_{j=1,2,\dots}$, defined by $\Theta_{j+1} = \Theta_j + k_j \nabla_{\Theta} \tilde{f}(\Theta_j)$ in (8), is convergent. The sequence $\{\Theta_j\}_{j=1,2,\dots}$ converges to a local maximum of $\{\tilde{f}_{Y|\mathbf{t}}(I(\mathbf{t})|\mathbf{t}, \Theta)\}$.

Proof. From (5),

$$\tilde{f}(\Theta_{j+1}) - \tilde{f}(\Theta_j) \geq C \sum_{i=1}^n w_{i,j} (\epsilon_{i,j}^c)^2 - (\epsilon_{i,j}^c - (\Theta_{j+1} - \Theta_j)^T (g(\mathbf{t}_i) - \overline{g(\mathbf{t})}))^2$$

Now, we take the next step in the direction of the gradient, i.e., we seek some $k_j > 0$ such that $\Theta_{j+1} = \Theta_j + k_j \nabla_{\Theta} \tilde{f}(\Theta_j)$ and k_j is the solution of,

$$k_j = \underset{k}{\operatorname{argmin}} \sum_{i=1}^n w_{i,j} (\epsilon_{i,j}^c - k \sum_{l=1}^n w_{l,j} \epsilon_{l,j}^c g^c(\mathbf{t}_l)^T g^c(\mathbf{t}_i))^2$$

Solving for k_j , we get,

$$k_j = \frac{\|\sum_{i=1}^n w_{i,j} \epsilon_{i,j}^c g(\mathbf{t}_i)\|^2}{\sum_{i=1}^n w_{i,j} \langle g(\mathbf{t}_i), \sum_{l=1}^n w_{l,j} \epsilon_{l,j}^c g(\mathbf{t}_l) \rangle^2} =: \frac{N(k_j)}{D(k_j)} \geq \frac{1}{\sum_{i=1}^n w_{i,j} \|g(\mathbf{t}_i)\|^2} > 0$$

Hence,

$$\begin{aligned} \text{RHS of (8)} &= C \sum_{i=1}^n w_{i,j} \left[\epsilon_{i,j}^c{}^2 - (\epsilon_{i,j}^c - k_j \sum_{l=1}^n w_{l,j} \epsilon_{l,j}^c g(\mathbf{t}_l)^T g(\mathbf{t}_i))^2 \right] \\ &= C \frac{N(k_j) \|\nabla_{\Theta} \tilde{f}(\Theta_j)\|^2}{D(k_j)} = C(\Theta_{j+1} - \Theta_j) \cdot \nabla_{\Theta} f(\Theta_j) = \frac{C}{k_j} \|\Theta_{j+1} - \Theta_j\|^2 \geq 0 \end{aligned}$$

This implies the convergence of $S_f \doteq \{\tilde{f}(\Theta_j)\}_j$ and that $\|\delta\Theta_j\| \rightarrow 0$ as in the proof for Theorem 4. The convergence of $S_{\Theta} \doteq \{\Theta_j\}_{j=1,2,\dots}$ to a local maximum similarly follows. \square

The gradient descent algorithm is more stable (albeit slower) since it does not require matrix inversion. Hence, we use this algorithm for applications in Section 5.

4 Empirical Validation

We tested the robustness of our algorithm empirically and compared its statistical performance with (a) the least squares (LS) method, and, (b) the least trimmed squares (LTS) method. The comparison to LS was done since it is the most popular parameter estimation framework apart from being the minimum-variance unbiased estimator (MVUE) for additive white Gaussian noise (AWGN), the most commonly used noise model. Further, our algorithm is a re-weighted LS (RLS) algorithm. LTS [19] was chosen since it is a state-of-the-art method for robust regression. We note here that LS is not robust while LTS has a positive breakdown value of $([(n-p)/2] + 1)/n$ [19]. In computer vision, one requires robust algorithms that have a higher breakdown value [20]. Redescending M-estimators (which the proposed formulation yields) are shown to have such a property. Thus, theoretically, they are more robust than either of the estimators described above. Nonetheless, it is useful to benchmark the performance of the proposed estimator with the above two estimators.

For ease of comparison, we used a 1-D data set as our test bed (Table 1). Intensity samples were generated using the equation $I_0(x) = ax^2 + bx + c$ at unit intervals from $x = -50$ to $x = 50$. To test the noise performance, we added uncorrelated noise from a noise distribution to each intensity sample independently. We used Gaussian and two-sided log-normal distributions as the noise density functions. We tested the performance at several values of the variance parameter associated with these distributions. At each value of variance, we generated 1000 realizations and calculated the bias and the variance of the two estimators. The Gaussian distribution was chosen as the LS estimator is the minimum variance unbiased estimator (MVUE) for AWGN noise while the log-normal distribution was chosen to simulate outliers. We present the results in Table 1: results for i.i.d. Gaussian noise are presented in the left column and for i.i.d. log-normal noise are presented in the right column. The ground-truth values are $a = 0.135$, $b = 0.55$ and $c = 1.9$. The results show that the performance of the proposed estimator is better than the LTS and the LS estimators when the number of outliers are large (log-normal noise). In case the noise is AWGN, standard deviation of the proposed estimator stays within twice that of the LS estimator (which is the MVUE). However, the proposed algorithm has a better breakdown value than either of the two estimators. Further, the proposed estimator is easy to implement as a recursive least squares (RLS) algorithm, and, in our simulations, it was an order of magnitude faster than the LTS algorithm.

5 Applications

The theoretical formulation derived in the previous sections provides a robust mechanism for parametric (global) or semi-parametric (local) image regression. This is achieved via a parametric representation of the image signal coupled with non-parametric kernel density estimation for the additive noise. As a consequence, the proposed estimation framework can be used for the following tasks:

Table 1. LS, LTS and KML estimates for (a, b, c) . $I_i = ax_i^2 + bx_i + c + \epsilon_i$. $\{\epsilon_i\}$ are i.i.d. Gaussian and i.i.d. log-normal with parameters $(0, \sigma^2)$ for experiments in left and right columns respectively. Ground-truth values are $(a, b, c) = (0.135, 0.55, 1.9)$. Mean and standard deviation of the estimated values for 1000 experiments are presented for each estimator - LS, LTS and KML, in rows 1, 2, and 3, respectively. KML performs better for log-normal while LTS and LS perform better for gaussian noise.

LS - Gaussian noise						
σ	Mean			Standard Deviation		
	a	b	c	a	b	c
1	0.135	0.550	1.907	0.0001	0.0036	0.1516
5	0.135	0.549	1.952	0.0007	0.0165	0.7474
9	0.135	0.549	1.853	0.0012	0.0308	1.3658
13	0.135	0.548	1.951	0.0017	0.0449	1.9375
(a)						
LTS - Gaussian noise						
λ	Mean			Standard Deviation		
	a	b	c	a	b	c
1	0.135	0.549	1.905	0.0002	0.0036	0.1881
5	0.135	0.547	2.005	0.0007	0.0177	0.8838
9	0.135	0.551	2.000	0.0011	0.0292	1.2843
13	0.135	0.550	2.003	0.0015	0.0335	1.7416
(c)						
KML - Gaussian noise						
σ	Mean			Standard Deviation		
	a	b	c	a	b	c
1	0.135	0.549	1.904	0.0002	0.0036	0.2697
5	0.135	0.547	1.960	0.0008	0.0166	1.0517
9	0.135	0.546	1.950	0.0015	0.0311	1.8904
13	0.135	0.544	1.951	0.0021	0.0481	2.6815
(e)						
LS - log-normal noise						
σ	Mean			Standard Deviation		
	a	b	c	a	b	c
1	0.135	0.550	1.917	0.0004	0.0098	0.4016
2	0.135	0.555	1.729	0.0063	0.1369	7.6999
3	0.140	0.194	6.673	0.2862	6.7165	2.36e2
4	0.762	1.605	-8.7e2	2.00e1	3.68e2	3.05e4
(b)						
LTS - log-normal noise						
λ	Mean			Standard Deviation		
	a	b	c	a	b	c
1	0.135	0.549	1.810	0.0005	0.0148	0.6224
2	0.135	0.550	1.996	0.0007	0.0179	0.8286
3	0.135	0.549	1.963	0.0009	0.0257	1.0629
4	0.135	0.548	1.973	0.0014	0.0380	1.7173
(d)						
KML - log-normal noise						
σ	Mean			Standard Deviation		
	a	b	c	a	b	c
1	0.135	0.549	1.931	0.0004	0.0083	0.4570
2	0.135	0.548	1.927	0.0007	0.0164	0.7446
3	0.135	0.548	1.916	0.0007	0.0198	0.8184
4	0.135	0.547	1.992	0.0011	0.0204	1.5543
(f)						

(A1) Edge preserving denoising of images that can be modelled (locally or globally) using the linear parametric model in Definition 1. (A2) Further, since the KML Estimation framework admits a multimodal error model, the framework can be used to partition data generated from a mixture of sources, each source generating data using the aforementioned parametric model (in such a case, we use $H_t \rightarrow \infty$. In other words, spatial kernel is assumed identically equal to one). Thus, the proposed framework provides a systematic way of doing Hough Transforms. Indeed, Hough Transform is the discretized version of $\tilde{f}_{Y|t}(\Theta^T g(t))$ if $H_Y \rightarrow 0$ and $H_t \rightarrow 0$.

To illustrate an application of the proposed estimation framework, we apply it to the task of range image segmentation. As test images, we use the database⁷ [21] generated using the ABW structured light camera. The objects imaged in

⁷ Range Image Databases provided on the University of South Florida website <http://marathon.csee.usf.edu/range/DataBase.html>.

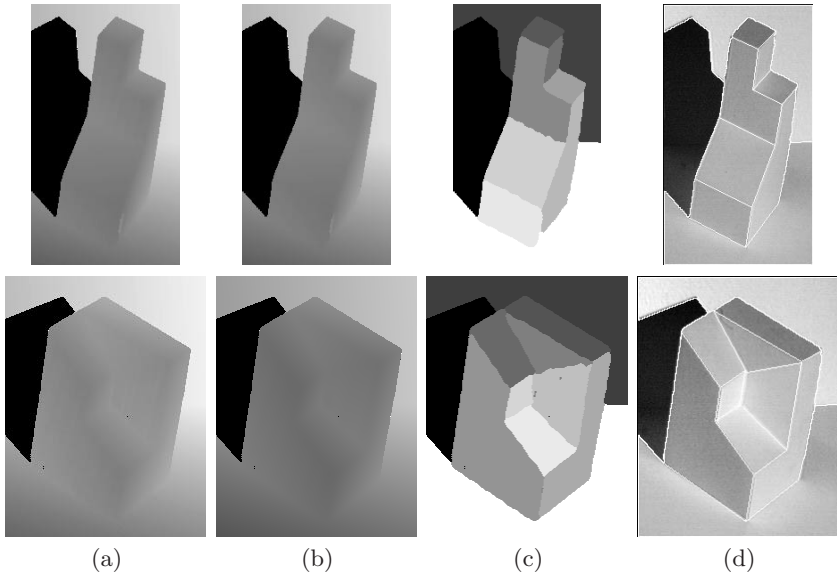


Fig. 2. Results for ABW test images 27 (row 1) and 29 (row 2). (a) Original range image, (b) reconstructed range image $\hat{I}(\mathbf{t})$ (locally planar assumption), (c) extracted planar segments, (d) Final segmentation after geometric boundary refinement. Edges are superimposed on the intensity images.

the database are all piecewise linear and provide a good testbed for the proposed estimation framework.

Let the range image be denoted by $Y(\mathbf{t})$. Then, for each \mathbf{t} in the image domain \mathcal{D} , the range may be represented as $Y(\mathbf{t}) = \langle [\mathbf{t}^T \mathbf{1}]^T, \Theta(\mathbf{t}) \rangle + \epsilon(\mathbf{t})$ where $\Theta(\mathbf{t}) \in S_\Theta \doteq \{\Theta_1, \dots, \Theta_M\}$ such that there are M unknown planar regions in the range image. The task of planar range image segmentation is to estimate the set S_Θ and to find the mapping, $\mathcal{T} : \mathcal{D} \rightarrow S_\Theta$.

The algorithm used for segmentation has following steps: (1) Estimate⁸ local parameters, $\tilde{\Theta}(\mathbf{t})$ for each \mathbf{t} . This also provides an edge-preserved de-noised range image ($\hat{I}(\mathbf{t})$). (2) Sort $\tilde{\Theta}(\mathbf{t})$'s in descending order of likelihood values, $\tilde{f}_{Y|\mathbf{t}}(\langle [\mathbf{t}^T \mathbf{1}]^T, \tilde{\Theta}(\mathbf{t}) \rangle)$. (3) Starting with the most likely $\tilde{\Theta}(\mathbf{t})$, estimate the parameters (say, $\tilde{\Theta}_p$) of the largest (supported) plane (spatial bandwidth matrix ∞ for global estimation). Since kernel maximum likelihood function can have several local minima, sorting has the effect of avoiding these local minima. Next, remove the data points supported by this plane ($\tilde{f}_{Y|\mathbf{t}}(\langle [\mathbf{t}^T \mathbf{1}]^T, \tilde{\Theta}_p(\mathbf{t}) \rangle)$ greater than a threshold) from the set of data points as well as from the sorted list. (4) Choose the next (remaining) point from the sorted list and fit the next largest plane to the remaining data points. (5) Iterate Step (4) N times ($N \gg M$, we choose $N = 100$) or until all data points get exhausted. (6) Discard the planes smaller than k pixels (resulting in N_1 planes),

⁸ KML estimation requires specification of the bandwidth parameters. These are hand-selected currently although they can be estimated akin to [17].

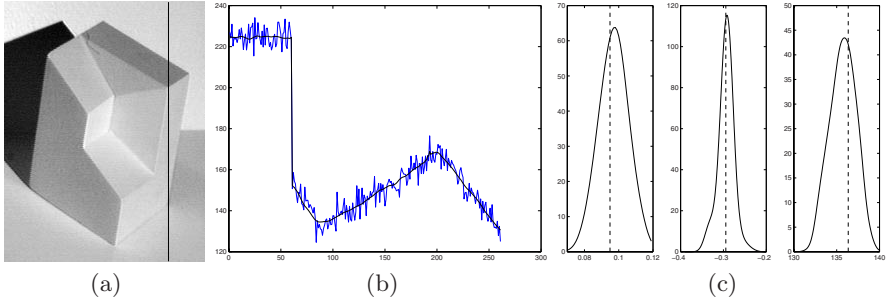


Fig. 3. (a) ABW test image 29. (b) Noisy and reconstructed range image (locally planar assumption) values are shown, along the scan line in (a), depicting robustness to discontinuities. (c) Estimated histogram of computed planar parameters for each data point of the leftmost object plane are shown. True image model : $I = 0.095y - 0.294x + 136.35$.

and construct the set $S_{\Theta} \doteq \{\tilde{\Theta}_1, \dots, \tilde{\Theta}_{N_1}\}$. (7) Reclassify all data points based on the likelihood values, i.e. the function $\mathcal{T}(\cdot)$ is initially estimated to be $\mathcal{T}(\mathbf{t}) = \operatorname{argmax}_{i \in 1 \dots N_1} \tilde{f}_{Y|\mathbf{t}}(\langle [\mathbf{t}^T \mathbf{1}]^T, \tilde{\Theta}_p(\mathbf{t}) \rangle | (\mathbf{t}), \tilde{\Theta}_i)$. (8) Repeat step (6). (9) Refine boundaries based on the geometric reasoning that all intersecting planes produce straight lines as edges (refer to Chapter 6, [18]).

In Figure 2, we show segmentation results on two⁹ range images from the ABW data set. The range values in column (a) properly belong to piecewise planar range data. The range data is missing in the (black) shadowed regions. We treat these regions as planar surfaces that occlude all neighboring surfaces. The estimated locally linear surface (output of step 1, $\hat{I}(\mathbf{t})$) is shown in column (b). To better illustrate the results of KML estimation, we reproduce a scan line in Figure 3(a). Note, in Figure 3(b), that occluding edge (discontinuity in the range data) is well preserved while denoising within continuous regions takes place as expected. This illustrates that the kernel maximum likelihood estimation preserves large discontinuities while estimating image parameters in a robust fashion. At the same time, locally estimated planar parameters are close to true values (refer to Figure 3(c) and the caption).

After planar regions are extracted in step (7), we show the results in Figure 2(c). Here, the brightness coding denotes different label numbers. We see that very few spurious regions are detected - this validates the robustness of global parameter estimation framework since we fit global planes to the data and iteratively remove data points supported by the planes. The goodness of the estimated plane parameters now allows us to carry out the geometric boundary refinement step as we can now estimate the corner and edge locations as detailed above. The final result after this is depicted in Figure 2(d). As one can see, the estimated segmentation is near perfect. To see that our results are better than several well-known range image segmentation algorithms, please compare them visually with those presented in Hoover et al [21].

⁹ More results will be included in the two extra pages allowed for the final manuscript.

6 Conclusions

In this paper, we presented a robust maximum likelihood parameter estimation framework by modelling the image with a linear parametric model and additive noise using kernel density estimators. The resulting estimator, the KML Estimator, is a redescending M-estimator. This novel approach provides a link between robust estimation of parametric image models and nonparametric density models for additive noise. We also provided a solution to the resultant nonlinear optimization problem and proved its convergence. Finally, we apply our framework to range image segmentation.

References

1. Huber, P.J.: Robust regression: Asymptotics, conjectures and monte carlo. *The Annals of Statistics* **1** (1973) 799–821
2. Yohai, V.J., Maronna, R.A.: Asymptotic behavior of M-estimators for the linear model. *The Annals of Statistics* **7** (1979) 258–268
3. Koenker, R., Portnoy, S.: M-estimation of multivariate regressions. *Journal of Amer. Stat. Assoc.* **85** (1990) 1060–1068
4. Chu, C.K., Glad, I.K., Godtliebsen, F., Marron, J.S.: Edge-preserving smoothers for image processing. *Journal of the American Statistical Association* **93** (1998) 526–541
5. Hillebrand, M., Muller, C.H.: On consistency of redescending M-kernel smoothers. unpublished manuscript - <http://www.member.uni-oldenburg.de/ch.mueller/publik/neuarticle5.pdf> **9** (2000) 1897–1913
6. Ayer, S., Sawhney, H.: Layered representation of motion video using robust maximum likelihood estimation of mixture models and MDL encoding. *Computer Vision, Int'l Conf.* (1995) 777–784
7. Black, M.J., Jepson, A.D.: Estimating optical flow in segmented images using variable-order parametric models with local deformations. *PAMI, IEEE Trans.* **18** (1996) 972–986
8. Kumar, R., Hanson, A.R.: Robust methods of estimating pose and a sensitivity analysis. *Image Understanding, CVGIP* (1994)
9. Liu, L., Schunck, B.G., Meyer, C.C.: On robust edge detection. *Robust Computer Vision, Int'l Workshop* (1990) 261–286
10. Mirza, M.J., Boyer, K.L.: Performance evaluation of a class of M-estimators for surface parameter estimation in noisy range data. *IEEE Trans., Robotics and Automation* **9** (1993) 75–85
11. Stewart, C.V.: Robust parameter estimation in computer vision. *SIAM Reviews* **41** (1999) 513–537
12. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. *Computer Vision, 6th Intl. Conf.* (1998) 839–846
13. Barash, D., Comaniciu, D.: A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. *Image and Video Computing* (2003)
14. Boorngaard, R., Weijer, J.: On the equivalence of local-mode finding, robust estimation and mean-shift analysis as used in early vision tasks. *Pattern Recognition, 16th Intl. Conf.* **3** (2002) 927–930

15. Hyndman, R.J., Bashtannyk, D.M., Grunwald, G.K.: Estimating and vizualizing conditional densities. *Journal of Comput. and Graph. Statistics* **5** (1996) 315–336
16. Chen, H., Meer, P.: Robust computer vision through kernel density estimation. *Computer Vision, 7th Euro. Conf.* **1** (2002) 236–250
17. Singh, M.K., Ahuja, N.: Regression-based bandwidth selection for segmentation using Parzen windows. *Computer Vision, 9th Intl. Conf.* (2003) 2–9
18. Singh, M.K.: Image Segmentation and Robust Estimation Using Parzen Windows. University of Illinois at Urbana-Champaign. Ph.D. Thesis (2003)
19. Rousseeuw, P.J., Van Driessen, K.: Computing LTS Regression for Large Data Sets. University of Antwerp. Technical Report (1999)
20. Mizera, I., Muller, C.H.: Breakdown points and variation exponents of robust M-estimators in linear models. *Annals of Statistics* **27** (1999) 1164–1177
21. Hoover, A., J.-B., G., Jiang, X., Flynn, P.J., Bunke, H., Goldgof, D.B., Bowyer, K.K., Eggert, D.W., Fitzgibbon, A.W., Fisher, R.B.: An experimental comparison of range image segmentation algorithms. *PAMI, IEEE Trans.* **18** (1996) 673–689

Keyframe Selection for Camera Motion and Structure Estimation from Multiple Views

Thorsten Thormählen, Hellward Broszio, and Axel Weissenfeld

Information Technology Laboratory, University of Hannover,
Schneiderberg 32, 30167 Hannover, Germany
{thormae, broszio, aweissen}@tnt.uni-hannover.de
<http://www.tnt.uni-hannover.de/~thormae>

Abstract. Estimation of camera motion and structure of rigid objects in the 3D world from multiple camera images by bundle adjustment is often performed by iterative minimization methods due to their low computational effort. These methods need a robust initialization in order to converge to the global minimum. In this paper a new criterion for keyframe selection is presented. While state of the art criteria just avoid degenerated camera motion configurations, the proposed criterion selects the keyframe pairing with the lowest expected estimation error of initial camera motion and object structure. The presented results show, that the convergence probability of bundle adjustment is significantly improved with the new criterion compared to the state of the art approaches.

1 Introduction

The estimation of camera motion and structure of rigid objects in the 3D world using camera images from multiple views has a long and sophisticated research history within the computer vision community.

Usually a mathematical parameter model of a pinhole camera with perspective projection is used to describe the mapping between the 3D world and the 2D camera image. To estimate the parameters of the camera model most approaches establish corresponding feature points in each view. By the introduction of a statistical error model, that describes the errors in the position of the detected feature points, a Maximum Likelihood estimator can be formulated that simultaneously estimates the camera parameters and the 3D positions of feature points. This joint optimization is called bundle adjustment [1].

If the errors in the positions of the detected feature points obey a Gaussian distribution, the Maximum Likelihood estimator has to minimize a nonlinear least squares cost function. In this case, fast minimization is carried out with iterative parameter minimization methods, like the sparse Levenberg-Marquardt method [1][2, Appendix 4.6].

The main difficulty of the iterative minimization is the robust initialization of the camera parameters and the 3D positions of feature points in order to converge

to the global minimum. One possible solution is to obtain an initial guess from two [3,4] or three [5,6] selected views out of the sequence or sub-sequence. These views are called keyframes.

Keyframes should be selected with care, for instance a sufficient baseline between the views is necessary to estimate initial 3D feature points by triangulation. Additionally, a large number of initial 3D feature points is desirable.

By comparison, keyframe selection has been neglected by the computer vision community. In the case of initialization from two views, Pollefeys et al. [3] use the Geometric Robust Information Criterion (GRIC) proposed by Torr [7]. This criterion allows to evaluate which model, homography (H-matrix) or epipolar geometry (F-matrix), fits better to a set of corresponding feature points in two view geometry. If the H-matrix model fits better than the F-matrix model, H-GRIC is smaller than F-GRIC and vice versa. For very small baselines between the views GRIC always prefers the H-matrix model. Thus, the baseline must exceed a certain value before F-GRIC becomes smaller than H-GRIC.

Pollefeys' approach searches for one keyframe pairing by considering all possible pairings of the first view with consecutive views in the sequence. Thus, the first keyframe of the keyframe pairing is always the first view of the sequence. The second keyframe is the last view for which the number of tracked feature points is above 90% of the number of feature points tracked at the view for which F-GRIC becomes smaller than H-GRIC. This approach guarants a certain baseline and a large number of initial 3D feature points.

Gibson et al. [4] propose a quite similar approach. Instead of GRIC they evaluate a score consisting of three weighted addends. The first addend becomes small if the number of reconstructed initial 3D feature points reduces in the actual keyframe pair compared to the previous keyframe pair. The second addend is the reciprocal value of the median reprojection error when a H-matrix is fitted to the feature points and the third addend is the median reprojection error when the F-matrix model is applied. Gibson's approach marks the pairing with the lowest score as keyframes.

The disadvantage of both approaches is, that they do not select the best possible solution. For instance, a keyframe pairing with a very large baseline is not valued better than a pairing with a baseline that just ensures that the F-matrix model fits better than the H-matrix model. Thus, only the degenerated configuration of a pure camera rotation between the keyframe pairing is avoided. Especially, if the errors in the positions of the detected feature points are high, these approaches may estimate a F-matrix, that does not represent the correct camera motion and therefore provides wrong initial parameters for the bundle adjustment.

The approach for keyframe selection presented in this paper formulates a new criterion using techniques from stochastic. By evaluating the lower bound for the resulting estimation error of initial camera parameters and initial 3D feature points, the keyframe pairing with the best initial values for bundle adjustment is selected. It will be shown that this new approach increases significantly the convergence probability of the bundle adjustment.

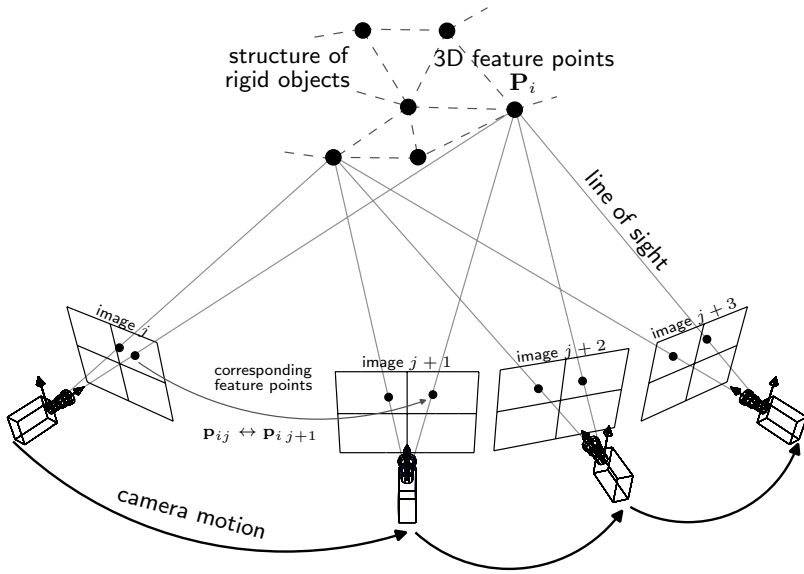


Fig. 1. Projection of 3D feature points on rigid objects in multiple camera views.

The following chapter defines a reference framework for the keyframe selection approaches by describing the processing steps that are used for the estimation of camera motion and structure of the observed objects. In Chapter 3 the new approach for keyframe selection is presented. Chapter 4 compares results of the different approaches in the defined framework and conclusions are drawn in Chapter 5.

2 Reference Framework

For estimation of camera motion parameters from corresponding feature points, the real camera must be represented by a mathematical camera model. The camera model describes the projection of a 3D feature point \mathbf{P} to the image coordinate \mathbf{p} through a perspective camera. Using homogeneous representation of coordinates, a 3D feature point is represented as $\mathbf{P} = (X, Y, Z, 1)^T$ and a 2D image feature point as $\mathbf{p} = (x, y, 1)^T$. Where \mathbf{p}_{ij} is the projection of a 3D feature point \mathbf{P}_i in the j -th camera (see Fig. 1), with

$$\mathbf{p}_{ij} \sim \mathbf{K}_j [\mathbf{R}_j | \mathbf{t}_j] \mathbf{P}_i = \mathbf{A}_j \mathbf{P}_i \quad \forall j \in \{1, \dots, J\}, i \in \{1, \dots, I\} \quad (1)$$

where \mathbf{K}_j is the calibration matrix, \mathbf{R}_j is the rotation matrix, \mathbf{t}_j is the translation vector, and \mathbf{A}_j is the camera matrix of the j -th camera. The software system used for estimation of \mathbf{A}_j and \mathbf{P}_i consists of five processing steps, as shown in Fig. 2. Each processing step is described briefly in the following subsections. Detailed related reading may be found in [2].

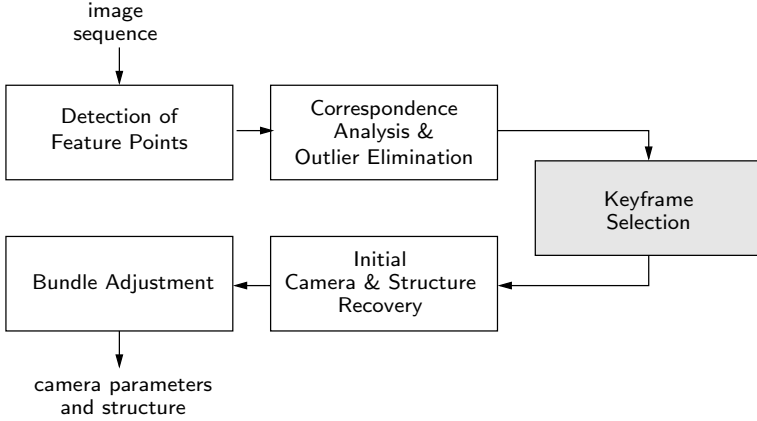


Fig. 2. Processing steps for the estimation of camera parameters and structure from image sequences

2.1 Detection of Feature Points

2D image feature points $\tilde{\mathbf{p}}$ are detected with sub-pixel accuracy using Harris' feature point detector [8]. For each image j of the sequence a list of feature point coordinates $L_j = \{\tilde{\mathbf{p}}_{1j}, \dots, \tilde{\mathbf{p}}_{ij}, \dots, \tilde{\mathbf{p}}_{Ij}\}$ is extracted. Due to noise in the intensity values of the images, the positions of the detected feature points $\tilde{\mathbf{p}} = (\tilde{x}, \tilde{y}, 1)^\top$ differ from the true positions $\mathbf{p} = (x, y, 1)^\top$, with

$$\tilde{x} = x + \Delta x \quad \text{and} \quad \tilde{y} = y + \Delta y \quad (2)$$

The error model in this paper assumes that Δx and Δy of all points $\tilde{\mathbf{p}}_{ij}$ are uncorrelated and obey a zero-mean Gaussian distribution with covariance matrix

$$\Sigma_{\tilde{\mathbf{p}}_{ij}} = \begin{pmatrix} \sigma_{x_{ij}}^2 & 0 \\ 0 & \sigma_{y_{ij}}^2 \end{pmatrix} \quad (3)$$

2.2 Correspondence Analysis and Outlier Elimination

The feature points in list L_j and L_{j+1} of two successive views are assigned by measuring normalized cross-correlation between 15×15 pixel windows surrounding the feature points. The correspondences are established for those feature points, which have the highest cross-correlation. This results in a list of correspondences $L_c = \{q_1, \dots, q_i, \dots, q_I\}$, where $q_i = (\tilde{\mathbf{p}}_{ij}, \tilde{\mathbf{p}}_{i,j+1})$ is a correspondence.

Due to erroneous assignment of feature points arising from moving objects, illumination changes or similarities in the scene, usually some of the correspondences are incorrect. Most of these outliers can be detected because they must fulfill the epipolar constraint between two views:

$$\mathbf{p}_{i,j+1}^\top \mathbf{F} \mathbf{p}_{ij} = 0 \quad \forall i \quad \text{and} \quad \det(\mathbf{F}) = 0 \quad (4)$$

where $\mathbf{F} = \mathbf{K}_{j+1}^{-\top} [\mathbf{t}_j]_x \mathbf{R} \mathbf{K}_j^{-1}$ is the F-matrix. In the case of motion degeneracy, if the camera does not translate between the views, or structure degeneracy, if the viewed scene structure is planar, a homography is the stricter constraint between two views:

$$\mathbf{p}_{i,j+1} = \mathbf{H} \mathbf{p}_{ij} \quad \forall i \quad (5)$$

where $\mathbf{H} = \mathbf{K}_{j+1} \mathbf{R} \mathbf{K}_j^{-1}$ is the H-matrix. \mathbf{H} or \mathbf{F} should be estimated by minimizing the residual error \bar{e} of the Maximum Likelihood cost function for the used error model, consequently here:

$$\bar{e}^2 = \frac{1}{4I} \sum_{i=1}^I d(\tilde{\mathbf{p}}_{ij}, \hat{\mathbf{p}}_{ij})_{\Sigma}^2 + d(\tilde{\mathbf{p}}_{i,j+1}, \hat{\mathbf{p}}_{i,j+1})_{\Sigma}^2 = \frac{1}{4I} \sum_{i=1}^I e_i^2 \longrightarrow \min \quad (6)$$

subject to $\hat{\mathbf{p}}_{ij}$ and $\hat{\mathbf{p}}_{i,j+1}$ fulfill exactly Eq. 4 for F-matrix estimation and Eq. 5 for the estimation of H-matrix, where $d(\dots)_{\Sigma}$ denotes the Mahalanobis distance for the given covariance matrices, here $\Sigma_{\tilde{\mathbf{p}}_{ij}}$ and $\Sigma_{\tilde{\mathbf{p}}_{i,j+1}}$. To achieve a robust estimation the random sampling algorithm MSAC (see [9,10] for details) is employed.

After estimation of \mathbf{H} and \mathbf{F} , Torr's GRIC is used to decide which of the both models should be used for outlier elimination and guided matching [7].

$$\text{GRIC} = \left(\sum_{i=1}^I \rho(e_i^2) \right) + \lambda_2 m I + \lambda_2 k \quad (7)$$

$$\text{with} \quad \rho(e^2) = \begin{cases} \frac{e^2}{\sigma^2} & \text{for } \frac{e^2}{\sigma^2} < \lambda_3(r-m) \\ \lambda_3(r-m) & \text{for } \frac{e^2}{\sigma^2} \geq \lambda_3(r-m) \end{cases} \quad (8)$$

where k is number of essential parameters of the model, m is dimension of the fitted manifold, and r is dimension of the measurements, with $k=7$, $m=3$, $r=4$ for F-GRIC and $k=8$, $m=2$, $r=4$ for H-GRIC. The model with the lower GRIC is indicated as more likely.

2.3 State of the Art in Keyframe Selection

In the keyframe selection step keyframe pairings are determined to start the initial camera and structure recovery in the following step.

In general, many possible keyframe pairings exist. To reduce complexity Pollefeys' and Gibson's approaches always set the first keyframe of a keyframe pairing at the first view. Then consecutive views of the sequence are considered. For comparability this procedure is also adopted in our reference framework.

Pollefeys' approach chooses as second keyframe the last view for which the number of tracked feature points is above 90% of the number of feature points tracked at the view where F-GRIC becomes smaller than H-GRIC.

In Gibson's approach the following score S_g is evaluated for each pairing of views:

$$S_g = w_1 \left(1.0 - \frac{I_1}{I_2} \right) + w_2 \frac{1}{\bar{e}_H^2} + w_3 \bar{e}_F^2 \quad (9)$$

where I_2 is the number of 3D feature points that were reconstructed in the previous pair and I_1 is the number of those features that can also be reconstructed in the currently evaluated pair, \bar{e}_H is the residual error defined in Eq. 6 with the H-matrix model fitted to the data, and \bar{e}_F is the residual error for the F-matrix model. The pairing with the lowest S_g is marked as new keyframe. Gibson suggests to choose the weights $w_1 = 3$, $w_2 = 10$, $w_3 = 1$.

Pollefeys and Gibson apply different optimization strategies in their bundle adjustment step. While Pollefeys' approach uses Incremental Bundle Adjustment, Gibson's approach uses Hierarchical Merging of Subsequences. In the incremental approach one keyframe pairing per sequence must be selected. In contrast, the hierarchical approach divides the sequence into subsequences according to the chosen keyframes. Thus, in this case one keyframe pairing per subsequence is available.

In order to compare the two state of the art approaches and the new approach, a common framework must be defined. In our reference framework the incremental approach is used in the bundle adjustment step. Hence, only one keyframe pairing per sequence is selected.

2.4 Initial Camera and Structure Recovery

After a keyframe pairing is selected a F-matrix between the keyframes is estimated by MSAC using Eq. 6 with Eq. 4 as cost function. The estimated F-matrix is decomposed to retrieve initial camera matrices $\hat{\mathbf{A}}_{k1}$ and $\hat{\mathbf{A}}_{k2}$ of both keyframes. Initial 3D feature points $\hat{\mathbf{P}}'_i$ are computed using triangulation (see [2, Chapter 11]). Now bundle adjustment between two views is performed by sparse Levenberg-Marquardt iteration using Eq. 6 subject to $\tilde{\mathbf{p}}_{i\ k1} = \hat{\mathbf{A}}_{k1}\hat{\mathbf{P}}'_i$ and $\tilde{\mathbf{p}}_{i\ k2} = \hat{\mathbf{A}}_{k2}\hat{\mathbf{P}}'_i$ as cost function. Initial camera matrices $\hat{\mathbf{A}}_j$, with $k_1 < j < k_2$, of the intermediate frames between the keyframes are estimated by camera resectioning. Therefore, the estimated 3D feature points $\hat{\mathbf{P}}'_i$ become measurements $\tilde{\mathbf{P}}'_i$ in this step. Assuming the errors mainly in $\tilde{\mathbf{P}}'_i$ and not in $\tilde{\mathbf{p}}_{ij}$ the following cost function must be minimized:

$$\bar{\mu}_{\text{res}}^2 = \frac{1}{3I} \sum_{i=1}^I d(\tilde{\mathbf{P}}'_i, \hat{\mathbf{P}}_i)_{\Sigma}^2 \longrightarrow \min \quad (10)$$

subject to $\tilde{\mathbf{p}}_{ij} = \hat{\mathbf{A}}_j\hat{\mathbf{P}}_i$ for all i , where $\bar{\mu}_{\text{res}}$ is the residual error of camera resectioning.

2.5 Bundle Adjustment

The final bundle adjustment step optimizes all cameras $\hat{\mathbf{A}}_j$ and all 3D feature points $\hat{\mathbf{P}}_i$ of the sequence by sparse Levenberg-Marquardt iteration, with

$$\bar{\nu}_{\text{res}}^2 = \frac{1}{2JI} \sum_{j=1}^J \sum_{i=1}^I d(\tilde{\mathbf{p}}_{ij}, \hat{\mathbf{A}}_j\hat{\mathbf{P}}_i)_{\Sigma}^2 \longrightarrow \min \quad (11)$$

where $\bar{\nu}_{\text{res}}$ is the residual error of bundle adjustment. The applied optimization strategy is Incremental Bundle Adjustment: First Eq. 11 is optimized for the keyframes and all intermediate views with the initial values determined in the previous step. Then the reconstructed 3D feature points are used for camera resectioning of the consecutive views. After each added view the 3D feature points are refined and extended and a new bundle adjustment is carried out until all cameras and all 3D feature points are optimized.

3 Keyframe Selection Algorithm

In this chapter the new approach for keyframe selection is presented. The approach attempts to find the keyframe pairing that minimize the estimation error of the following final bundle adjustment step. Bundle adjustment with iterative minimization is heavily reliant on good initial values for 3D feature points $\hat{\mathbf{P}}_i$ and camera matrices $\hat{\mathbf{A}}_j$. Thus, a keyframe selection criterion that judges the quality of these initial values is needed. Therefore, it should be taken into account, that initial camera matrices $\hat{\mathbf{A}}_j$ are estimated from initial 3D feature points $\hat{\mathbf{P}}'_i$ as described in step 2.4, which rely on the choice of the keyframe pairing.

Consequently, the first step of the approach is the estimation of the covariance matrix of initial 3D feature points $\hat{\mathbf{P}}'_i$ for each keyframe pairing where F-GRIC is smaller than H-GRIC. If $\text{F-GRIC} \geq \text{H-GRIC}$ the keyframe pairing candidate is rejected without evaluation of the covariance matrix. In the second step the estimated covariance matrix is applied to approximate a lower bound for the estimation error of 3D feature points $\hat{\mathbf{P}}_i$ and camera matrices $\hat{\mathbf{A}}_j$ after camera resectioning.

3.1 Covariance Matrix Estimation

For the estimation of covariance matrix of initial 3D feature points $\hat{\mathbf{P}}'_i$ a bundle adjustment between the two analyzed keyframes with camera matrices \mathbf{A}_{k1} and \mathbf{A}_{k2} is performed. As derived in [2, Chapter 4.2], the covariance matrix $\Sigma_{\hat{\mathbf{A}}_k \hat{\mathbf{P}}'_i}$ of both cameras and the 3D feature points is the first order equal to:

$$\Sigma_{\hat{\mathbf{A}}_k \hat{\mathbf{P}}'_i} = (\mathbf{J}^\top \Sigma_{\hat{\mathbf{P}}}^{-1} \mathbf{J})^+ \quad (12)$$

where \mathbf{J} is the Jacobian matrix calculated at the optimum for $\hat{\mathbf{A}}_{k1}$, $\hat{\mathbf{A}}_{k2}$, and $\hat{\mathbf{P}}'_i$, and where $\Sigma_{\hat{\mathbf{P}}} = \text{diag}(\dots, \Sigma_{\hat{\mathbf{P}}_{ij}}, \dots)$ and $(\dots)^+$ denotes the pseudo-inverse. It should be stressed, that the bundle adjustment between two views and the covariance matrix $\Sigma_{\hat{\mathbf{A}}_k \hat{\mathbf{P}}'_i}$ can be estimated with significant time savings using techniques for sparse matrices, because $\mathbf{J}^\top \Sigma_{\hat{\mathbf{P}}}^{-1} \mathbf{J}$ has a sparse block structure (see [2, Appendix 4.6] for details). By extracting $\Sigma_{\hat{\mathbf{P}}'_i}$ from $\Sigma_{\hat{\mathbf{A}}_k \hat{\mathbf{P}}'_i}$ the total variance of the 3D feature points $\hat{\mathbf{P}}'_i$ is calculated by the trace of $\Sigma_{\hat{\mathbf{P}}'_i}$.

$$E \left[\sum_{i=1}^I d(\mathbf{P}_i, \hat{\mathbf{P}}'_i)^2 \right] = \text{trace}(\Sigma_{\hat{\mathbf{P}}'_i}) \quad (13)$$

where \mathbf{P}_i are the true 3D feature points, $E[\dots]$ denotes the expectation of the function, and $d(\dots)$ denotes the Euclidian distance.

3.2 Expectation of Estimation Error

Now, a lower bound for the mean estimation error $\bar{\mu}_{\text{est}}$ of 3D feature points $\hat{\mathbf{P}}_i$ and camera matrices $\hat{\mathbf{A}}_j$ after camera resectioning is derived (compare Eq. 10):

$$E[\bar{\mu}_{\text{est}}^2] = E \left[\frac{1}{3I} \sum_{i=1}^I d(\mathbf{P}_i, \hat{\mathbf{P}}_i)^2 \right] \quad (14)$$

The measurements in the cost function defined by Eq. 10 are the 3D feature points $\tilde{\mathbf{P}}'_i$, that correspond to the estimated $\hat{\mathbf{P}}'_i$ in the previous step. The estimated 3D feature points $\hat{\mathbf{P}}'_i$ obey a Gaussian distribution defined on a space of dimension $3I$. In order to simplify calculation and reduce computational effort, let us assume $\tilde{\mathbf{P}}'_i$ obey an isotropic Gaussian distribution with $\Sigma_{\tilde{\mathbf{P}}'_i} = \bar{\sigma}^2 \mathbf{I}$, where \mathbf{I} is the Identity matrix and $\bar{\sigma}^2 = \text{trace}(\Sigma_{\tilde{\mathbf{P}}'_i})/3I$, so that $\text{trace}(\Sigma_{\tilde{\mathbf{P}}'_i}) = \text{trace}(\Sigma_{\hat{\mathbf{P}}'_i})$.

The constraint $\tilde{\mathbf{p}}_{ij} = \hat{\mathbf{A}}_j \hat{\mathbf{P}}_i$ of Eq. 10 enforces that $\hat{\mathbf{P}}_i$ can be located only on the line of sight defined by $\tilde{\mathbf{p}}_{ij}$ and $\hat{\mathbf{A}}_j$. Thus, the degrees of freedom for every estimated 3D feature point $\hat{\mathbf{P}}_i$ reduces from three to one. This means, that within the measurement space of dimension $3I$ a surface of dimension $I+A$ exists, where A is the number of essential parameters of one camera \mathbf{A}_j . On this surface all possible solutions for $\hat{\mathbf{P}}_i$ and $\hat{\mathbf{A}}_j$ are located. In the Levenberg-Marquardt algorithm this solution surface is approximated by a tangent surface, which has the same dimension. Because the Gaussian distribution in the measurement space is assumed isotropic and thus invariant to rotation, the projection on the tangent surface is equal to the projection on the first $(I+A)$ coordinate axes of the measurement space (see [2, Chapter 4.1.3] for details). Thus, on the tangent surface one gets an isotropic Gaussian distribution with total variance $(I+A)\bar{\sigma}^2$. This results in a expected estimation error

$$E[\bar{\mu}_{\text{est}}^2] = \frac{1}{3I} (I+A) \bar{\sigma}^2 = \frac{I+A}{(3I)^2} \text{trace}(\Sigma_{\tilde{\mathbf{P}}'_i}) = S_c \quad (15)$$

If $\text{F-GRIC} < \text{H-GRIC}$, the score S_c is the new criterion to evaluate a keyframe pairing candidate, where the pairing with a lower S_c indicates a better choice. This is conceivable as a small $\text{trace}(\Sigma_{\tilde{\mathbf{P}}'_i})/(3I)$ corresponds to a small variances of the estimated initial 3D feature points and the quotient $(I+A)/(3I)$ becomes smaller for a larger number of 3D feature points I .

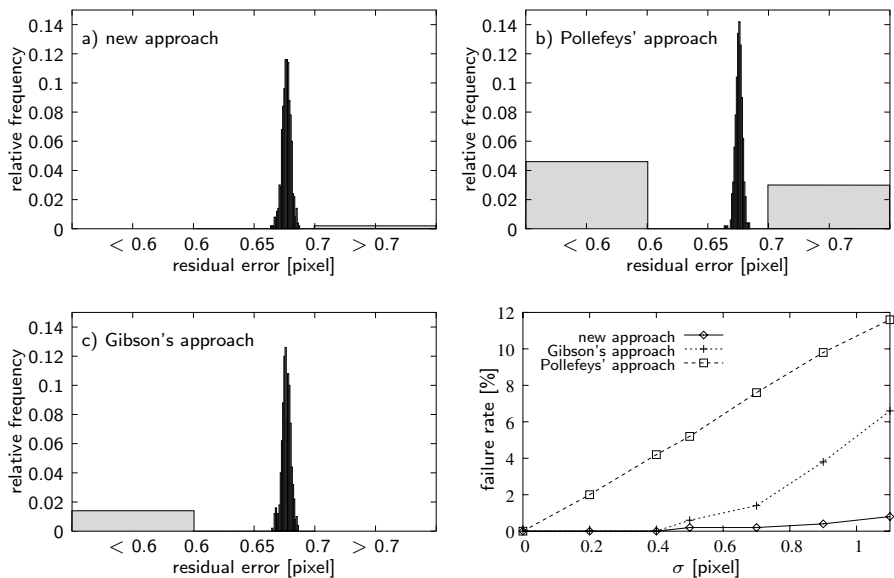


Fig. 3. Relative frequency of residual error $\bar{\nu}_{\text{res}}$ (see Eq. 11) for 500 trials. The errors in the positions of feature points obey a Gaussian distribution with standard deviation $\sigma = 0.7$ pixel. The left most and right most bins capture all residual errors that are smaller than 0.6 pixel and larger than 0.7 pixel. Trials with these residual errors are counted as failures of the bundle adjustment: a) new approach, b) Pollefeys' approach, and c) Gibson's approach. d) Failure rate over standard deviation σ for the different approaches.

4 Results

4.1 Synthetic Data Experiments

This chapter compares the new criterion with the state of the art approaches by Pollefeys and Gibson. Therefore, synthetic data experiments are carried out in the defined reference framework of Chapter 2, whereby only the keyframe selection criterion is changed.

500 synthetic test sequences with random camera motion and random scenes are generated. Each test sequence consists of 40 views. The camera translation between two successive views is uniformly distributed between 0 and 80 mm in all three coordinate directions and the camera rotation around the coordinate axes is uniformly distributed between 0 and 1 degree. 50% of the generated camera motions between two images are purely rotational. The camera has an image size of 720×576 pixel = 7.68×5.76 mm and a mean focal length of 10.74 mm. The random scenes consist of 4000 3D feature points, which have a distance from the camera between 800 and 3200 mm. Approximately 35 to 40 of these 3D feature points are used in the final bundle adjustment step. The errors in the positions of

the generated 2D image feature points obey an isotropic Gaussian distribution. 20% of the generated correspondences between 2D feature points are outliers.

Fig. 3a-c opposes the relative frequency of residual error $\bar{\nu}_{\text{res}}$ after bundle adjustment for the three approaches. 500 trials are performed and the errors in the feature point positions have a standard deviation $\sigma = 0.7$ pixel. The expectation value of the residual error

$$E[\bar{\nu}_{\text{res}}] = \sigma \sqrt{1 - \frac{AJ + 3I}{2JI}} \quad (16)$$

is approximately 0.65 pixel. Therefore, if a residual error is smaller than 0.6 pixel or larger than 0.7 pixel, the bundle adjustment has not converged to the correct minimum and these trials are counted as failures. It is obvious, that the new approach improves the convergence probability of the bundle adjustment significantly because failures occur less frequently. In Fig. 3d the failure rates over standard deviation σ for the different approaches are plotted. Especially, if the standard deviation is large, the new approach shows its improved robustness.



Fig. 4. Examples of augmented image sequences.

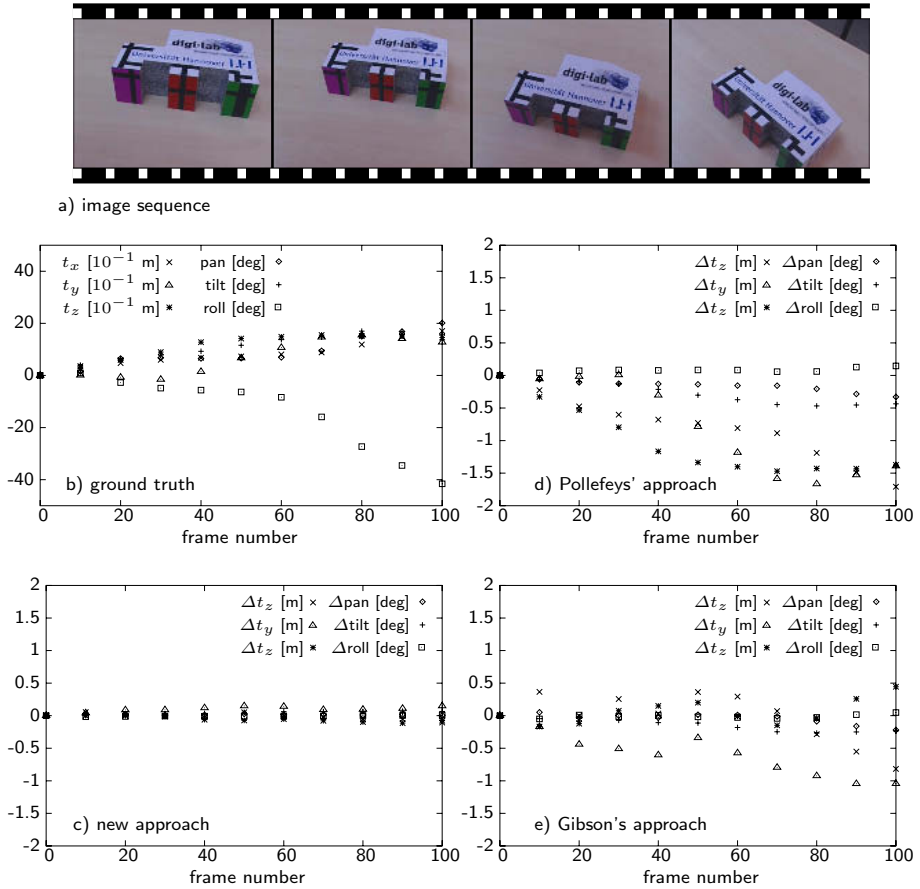


Fig. 5. a) Image sequence showing a test object with an exactly known structure. b) Ground truth extrinsic camera parameter generated with Tsai's camera calibration method. Difference between ground truth and estimation c) new approach, d) Pollefeys' approach e) Gibson's approach.

4.2 Natural Image Sequences

The new keyframe selection criterion has also demonstrated to work well on natural image sequences taken by a moving camera. Results of augmented image sequences that have been calibrated using the technique described in this paper are illustrated in Fig. 4. Videos of these augmented image sequences can be found on our website¹.

An empirical comparison of the new criterion with the state of the art approaches for natural image sequences is difficult because a large database containing natural image sequences with ground truth camera parameters would

¹ <http://www.digilab.uni-hannover.de/results.html>

be necessary. However, in order to illustrate the practical relevance of the new approach, a real-world example is given in Fig. 5. In Fig. 5a the evaluated image sequence is shown, which contains a test object with exactly known structure. Camera parameters are generated for every 10th view of this sequence with Tsai's camera calibration method [11], whereby the necessary 3D \leftrightarrow 2D correspondences are manually edited. Generated camera parameters are exhibited in Fig. 5b and serve as ground truth. In Fig. 5c-e the differences between the ground truth and the estimated camera parameters after bundle adjustment for the different keyframe selection approaches are plotted. In this example the bundle adjustment does not converge to the right solution, if Pollefeys' or Gibson's approach is used. In contrast, the new keyframe selection approach gives satisfying results. It should be stressed, that this single selected example gives no information about the general performance of the three keyframe selection criteria. However, this example reveals, that failures due to wrong keyframe selection can be observed not only in synthetic data experiments but also occur in practice.

5 Conclusion

A new criterion for keyframe selection is proposed. It is derived from the estimation of the covariance matrix of initial 3D feature points and a lower bound for the estimation error of camera resectioning. While the state of the art approaches just avoid degenerated camera motion configurations, the new approach searches for the best possible keyframe pairing. This results in more accurate initial values for 3D feature points and camera parameters. Thus, iterative parameter minimization methods that are applied in the bundle adjustment, like the sparse Levenberg-Marquardt method, converge more frequently into the global minimum.

Furthermore, we see no reason against an adaptation of the new criterion into the three view framework of [5,6], where the trifocal tensor is used and initial 3D feature points are estimated from three views. Though a verification of this is left for future work.

References

1. Triggs, B., McLauchlan, P., Hartley, R.I., Fitzgibbon, A.: Bundle adjustment – A modern synthesis. In: Workshop on Vision Algorithms. Volume 1883 of Lecture Notes in Computer Science. (2000)
2. Hartley, R.I., Zisserman, A.: Multiple View Geometry. Cambridge University Press (2000)
3. Pollefeys, M., Gool, L.V., Vergauwen, M., Cornelis, K., Verbiest, F., Tops, J.: Video-to-3d. In: Proceedings of Photogrammetric Computer Vision 2002 (ISPRS Commission III Symposium), International Archive of Photogrammetry and Remote Sensing. Volume 34. (2002) 252–258
4. Gibson, S., Cook, J., Howard, T., Hubbard, R., Oram, D.: Accurate camera calibration for off-line, video-based augmented reality. In: IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002), Darmstadt, Germany (2002)

5. Fitzgibbon, A., Zisserman, A.: Automatic camera recovery for closed or open image sequences. In: European Conference on Computer Vision. Volume 1406 of Lecture Notes in Computer Science. (1998) 311–326
6. Georgescu, B., Meer, P.: Balanced recovery of 3d structure and camera motion from uncalibrated image sequences. In: European Conference on Computer Vision. Volume 2351 of Lecture Notes in Computer Science. (2002) 294–308
7. Torr, P., Fitzgibbon, A., Zisserman, A.: The problem of degeneracy in structure and motion recovery from uncalibrated images. *International Journal of Computer Vision* **32** (1999) 27–44
8. Harris, C., Stephens, M.: A combined corner and edge detector. In: 4th Alvey Vision Conference. (1988) 147–151
9. Fischler, R.M.A., Bolles, C.: Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM* **24** (1981) 381–395
10. Torr, P.H.S., Zisserman, A.: MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding* **78** (2000) 138–156
11. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3-d machine vision metrology using off-the-shelf cameras and lenses. *IEEE Transaction on Robotics and Automation* **3** (1987) 323–344

Omnidirectional Vision: Unified Model Using Conformal Geometry

Eduardo Bayro-Corrochano and Carlos López-Franco

Cinvestav, Unidad Guadalajara, Computer Science Department,
GEOVIS LABORATORY, Jalisco, México
{edb,clopez}@gdl.cinvestav.mx, <http://www.gdl.cinvestav.mx/~edb>

Abstract. It has been proven that a catadioptric projection can be modeled by an equivalent spherical projection. In this paper we present an extension and improvement of those ideas using the conformal geometric algebra, a modern framework for the projective space of hyperspheres. Using this mathematical system, the analysis of diverse catadioptric mirrors becomes transparent and computationally simpler. As a result, the algebraic burden is reduced, allowing the user to work in a much more effective framework for the development of algorithms for omnidirectional vision. This paper includes complementary experimental analysis related to omnidirectional vision guided robot navigation.

1 Introduction

Living beings inhabit complex environments. In order to survive in these environments, they should be able to perceive the surrounding objects. One of the most important senses for object perception is vision. This sense is characterized by the ability to focus in a particular object with high precision, but it is also capable of simultaneously observing most of the changing surrounding medium.

In the case of robotic navigation it would be convenient if the robot could have a wide field of vision; but the traditional cameras are limited since they have a narrow field of view. This is a problem that has to be overcome to ease robotic navigation.

An effective way to increase the visual field is the use of a catadioptric sensor which consists of a conventional camera and a convex mirror. In order to be able to model the catadioptric sensor geometrically, it must satisfy the restriction that all the measurements of light intensity pass through only one point in the space (fixed view-point). The complete class of mirrors that satisfy this restriction were analyzed by Baker and Nayar [1].

To model the catadioptric sensor we can use an equivalent spherical projection defined by Geyer and Daniilidis [3]. In this paper, we present a new proposal of the spherical projection using conformal geometric algebra. The advantage of this algebra is that this system has the sphere as the basis geometric object and all the other objects are defined in terms of it (e.g., the intersections between entities can be computed using the *meet* operator). Hence we obtain a more transparent and compact representation due to the high-level symbolic

language of geometric algebra. In this new representation the user can compute and derive conclusion much easier. As a result the development of algorithms for omnidirectional vision becomes simpler and effective.

2 Unified Model

Recently, Geyer and Daniilidis [3] presented a unified theory for all the catadioptric systems with an effective viewpoint. They show nicely that these systems (parabolic, hyperbolic, elliptic) can be modeled with a projection through the sphere. In their paper they define a new notation where $A \vee B$ denotes the line joining the points A and B , and $l \wedge m$ denotes the intersection of the lines l and m . Also, this operator is used to denote the intersection of the line l and the conic c in the form $l \wedge m$ (note that this can result in a point pair). When the intersection is a point pair they distribute over the \vee, \wedge ; for example $A \vee (l \wedge c)$ is the pair $(A \vee P_1, A \vee P_2)$, where $P_{1,2}$ are points obtained from the intersection of l and c .

Definition of a quadratic projection. Let c be a conic, A and B two arbitrary points, ℓ any line not containing B and P a point in the space. The intersection of the line and the conic is a point pair R_1 and R_2 (possibly imaginary). The quadratic projection is defined as

$$Pq(c, A, B, \ell) \rightarrow (((P \vee A) \wedge c) \vee B) \wedge \ell. \quad (1)$$

Definition of a catadioptric projection. This projection is defined in terms of a quadratic projection where the points A and B are the focus F_1, F_2 of the conic c respectively, and ℓ is a line perpendicular to $F_1 \vee F_2$. The catadioptric projection is defined as

$$Pq(c, F_1, F_2, \ell) \rightarrow (((P \vee F_1) \wedge c) \vee F_2) \wedge \ell. \quad (2)$$

The important question now is: given a catadioptric projection with parameters (c, F_1, F_2, ℓ) , which are the parameters (c', A, B, ℓ') that result in an equivalent quadratic projection? This is not answered in general, but in a more restricted form we can ask ourselves: are there any parameters (c', A, B, ℓ') , where c' is a circle with a unit radius and center in A , B is some point and $\ell \parallel \ell'$, that produce an equivalent projection? To obtain equivalent projections they must have the same effective viewpoint and therefore $A = F_1$. Thus it is required to find ℓ' and B such that

$$q = (c, F_1, F_2, \ell) = q(c', F_1, B, \ell'). \quad (3)$$

Derivation of $q(c, F_1, F_2, \ell)$. The quadratic form of c in terms of its eccentricity ϵ and a scaling parameter $\lambda > 0$ is given by the equation

$$Q_{\epsilon, \lambda} = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 - 4\epsilon^2 & -4\epsilon\lambda \\ 0 & -4\epsilon\lambda & -4\lambda \end{pmatrix}. \quad (4)$$

Hence $F_1 = (0, 0, 1)$ and $F_2 = (0, -2\epsilon, \lambda^{-1}(\epsilon^2 - 1))$ are the foci of c with *latus rectum* 2λ , assuming that the intersection of the line ℓ and the y -axis is μ , so the line has the coordinates $[0, 1, -\mu]$. The first part of the catadioptric projection of a point P is $(P \vee F_1) \wedge c$, and can be expressed as

$$R_i = F_1 + \theta_i P, \quad (5)$$

where θ_i are the roots of the quadratic equation

$$0 = R_i Q_{\epsilon, \lambda} R_i^T = F_1 Q_{\epsilon, \lambda} F_1^T + 2\theta_i F_1 Q_{\epsilon, \lambda} P^T + \theta_i^2 P Q_{\epsilon, \lambda} P^T, \quad (6)$$

$$\theta_i = \frac{\lambda}{(-1)^i \sqrt{x^2 + y^2 - \epsilon y - \lambda w}}, \quad (7)$$

Later, the projection of the points R_i to the line $\ell = [0, 1, -\mu]$ from the point F_2 is

$$T_{\epsilon, \lambda, \mu} = \begin{pmatrix} -2\epsilon\lambda + \mu(1 - \epsilon^2) & 0 \\ 0 & 1 - \epsilon^2 \\ 0 & -2\epsilon\lambda \end{pmatrix}. \quad (8)$$

Finally the projected points Q_i are

$$Q_i = R_i T_{\epsilon, \lambda, \mu} = \left(x(2\epsilon\lambda - \mu(1 - \epsilon^2)), -(1 + \epsilon^2)y - 2(-1)^i \epsilon \sqrt{x^2 + y^2} \right). \quad (9)$$

Now, the projection $q(c', A, B, \ell')$ is the spherical projection — or in the transversal section, the projection to the circle. Let c' be a unit circle centered in F . The points R'_i are the intersections of the line $F_1 \vee P$ with this circle, and can be found by

$$R'_i = (-1)^i \sqrt{x^2 + y^2}, \quad (10)$$

the projection of the points R'_i to the line ℓ' is

$$U_{l, m} = \begin{pmatrix} l - m & 0 \\ 0 & -1 \\ 0 & l \end{pmatrix}, \quad (11)$$

and finally the projected points are

$$Q'_i = R'_i U_{l, m} = ((l - m)x, -y + l(-1)^i \sqrt{x^2 + y^2}). \quad (12)$$

Once the catadioptric and spherical projections have been calculated, the question it arises is for which B and ℓ' are $q(c, F_1, F_2, \ell') = q(c', F_1, F_2, \ell')$? If l and m can be chosen freely, independent of x, y, w then (9) and (12) are the same up to scale factor. The projections are equivalent if we choose

$$l = \frac{2\epsilon}{1 + \epsilon^2}, \quad (13)$$

$$m = \frac{\mu - \epsilon(\epsilon\mu + 2\lambda - 2)}{1 + \epsilon^2}. \quad (14)$$

Interesting enough with (13) and (14) we can model any catadioptric projection through the spherical projection, it is just a matter of calculating the parameters l and m according to the eccentricity and scaling parameters of the mirror. Next section outlines a brief introduction into the mathematical system which will help us to handle effectively omnidirectional vision in a new framework.

3 Geometric Algebra

The mathematical model used in this work is the geometric algebra (GA). This algebra is based on the Clifford and Grassmann algebras and the form used is the one developed by David Hestenes since late sixties [5].

In the n -dimensional geometric algebra we have the standard interior product which takes two vectors and produces a scalar, furthermore we have the wedge (exterior) product which takes two vectors and produces a new quantity that we call bivector or oriented area. Similarly, the wedge product of three vectors produces a trivector or oriented volume. Thus, the algebra has basic elements that are geometric oriented objects of different grade. The object with highest grade is called *pseudo-scalar* with the unit pseudo-scalar denoted by I (e.g. in 3D the unit pseudo-scalar is $e_1 \wedge e_2 \wedge e_3$). The outer product of r vectors is called an *r -blade* of grade r . A multivector is a quantity created by a linear combination of r -blades. Also, we have the geometric product which is defined for any multivector. The geometric product of two vectors is defined by the inner and outer product as

$$ab = a \cdot b + a \wedge b \quad (15)$$

Where the interior product (\cdot) and wedge product (\wedge) of two vectors $a, b \in \langle G_{p,q} \rangle_1 \equiv R^{p+q}$ can be expressed as

$$a \cdot b = \frac{1}{2}(ab + ba) \quad \text{and} \quad (16)$$

$$a \wedge b = \frac{1}{2}(ab - ba) . \quad (17)$$

As an extension, the interior product of an r -blade $a_1 \wedge \dots \wedge a_r$ with an s -blade $b_1 \wedge \dots \wedge b_s$ can be expressed as

$$\begin{aligned} & (a_1 \wedge \dots \wedge a_r) \cdot (b_1 \wedge \dots \wedge b_s) = \\ & \begin{cases} ((a_1 \wedge \dots \wedge a_r) \cdot b_1) \cdot (b_2 \wedge \dots \wedge b_s) & (\text{if } r \geq s) , \\ (a_1 \wedge \dots \wedge a_{r-1}) \cdot (a_r \cdot (b_1 \wedge \dots \wedge b_s)) & (\text{if } r < s) \end{cases} \end{aligned} \quad (18)$$

with

$$(a_1 \wedge \dots \wedge a_r) \cdot b_1 = \sum_{i=1}^r (-1)^{r-i} a_1 \wedge \dots \wedge a_{i-1} \wedge (a_i \cdot b_1) \wedge a_{i+1} \wedge \dots \wedge a_r ,$$

$$a_r \cdot (b_r \wedge \dots \wedge b_s) = \sum_{i=1}^s (-1)^{i-1} b_1 \wedge \dots \wedge b_{i-1} \wedge (a_r \cdot b_i) \wedge b_{i+1} \wedge \dots \wedge b_s . \quad (19)$$

The dual X^* of an r -blade X is

$$X^* = XI^{-1} . \quad (20)$$

The *shuffle* product $A \vee B$ satisfies the “DeMorgan rule”

$$(A \vee B)^* = A^* \wedge B^* . \quad (21)$$

3.1 Conformal Geometric Algebra

For a long time it has been known that using a projective description of the Euclidean 3D space in 4D has many advantages, particularly when the intersection of lines and planes are needed. Recently, these ideas were re-taken mainly by Hestenes [5], where he represents the Euclidean 3D space by a conformal space of 5D. In this conformal space the projective geometry is included, but in addition it can be extended to circles and spheres.

The real vector space $R^{n,1}$ or $R^{1,n}$ is called the Minkowski space, after the introduction of Minkowski’s space-time model in $R^{3,1}$. The Minkowski plane $R^{1,1}$ has the orthonormal basis $\{e_+, e_-\}$ defined by the properties

$$e_+^2 = 1, \quad e_-^2 = -1, \quad e_+ \cdot e_- = 0 . \quad (22)$$

Furthermore, the basis null vectors are

$$e_0 = \frac{1}{2}(e_- - e_+), \quad \text{and} \quad e = e_- + e_+, \quad (23)$$

with properties

$$e_0^2 = e^2 = 0, \quad e \cdot e_0 = -1 . \quad (24)$$

We will be working in the $R^{n+1,1}$ space, which can be decomposed in

$$R^{n+1,1} = R^n \oplus R^{1,1} . \quad (25)$$

This decomposition is known as the conformal split. Therefore any vector $a \in R^{n+1,1}$ admits the split

$$a = \mathbf{a} + \alpha e_0 + \beta e . \quad (26)$$

The conformal vector space derived from R^3 is denoted as $R^{4,1}$, its bases are $\{e_1, e_2, e_3, e_+, e_-\}$. The unit conformal pseudo-scalar is denoted as

$$I_c = e_{+-123} . \quad (27)$$

In the conformal space the basis entities are spheres

$$s = \mathbf{p} + \frac{1}{2}(\mathbf{p}^2 - \rho^2)e + e_0 . \quad (28)$$

A point x is nothing more than a sphere with radius $\rho = 0$, yielding

$$x = \mathbf{x} + \frac{1}{2}\mathbf{x}^2 e + e_0 . \quad (29)$$

The dual form s^* of a sphere s has the advantage that it can be calculated with four points in the sphere

$$s^* = a \wedge b \wedge c \wedge d . \quad (30)$$

The definition of the entities, its dual representation and its grade is shown in the Table 3.1.

Table 1. Entities in conformal geometric algebra

Entity	Representation	Grade	Dual Representation	Grade
Sphere	$s = \mathbf{p} + \frac{1}{2}(\mathbf{p}^2 - \rho^2)e + e_0$	1	$s^* = a \wedge b \wedge c \wedge d$	4
Point	$x = \mathbf{x} + \frac{1}{2}\mathbf{x}^2e + e_0$	1	$x^* = (-Ex - \frac{1}{2}x^2e + e_0)I_E$	4
Plane	$P = nI_E - de$ $n = (a - b) \wedge (a - c)$ $d = (a \wedge b \wedge c)I_E$	1	$P^* = e \wedge a \wedge b \wedge c$	4
Line	$L = rI_E - emI_E$ $r = (a - b)$ $d = (a \wedge b)$	2	$L^* = e \wedge a \wedge b$	3
Circle	$z = s_1 \wedge s_2$	2	$z^* = a \wedge b \wedge c$	3
Point Pair	$PP = s_1 \wedge s_2 \wedge s_3$	3	$PP^* = a \wedge b, X^* = e \wedge x$	2

3.2 Rigid Motion in the Conformal Geometric Algebra

In $G_{4,1}$ the rotations are represented by rotors $R = \exp(\frac{\theta}{2}l)$. Where the bivector l is the screw of rotation axis and the amount of rotation is given by the angle θ . An entity can be rotated by multiplying from the left with the rotor R and from the right with its reverse \tilde{R} (e.g., $x' = Rx\tilde{R}$).

One entity can be translated with respect to a translation vector t using the translator $T = 1 + \frac{et}{2} = \exp(\frac{et}{2})$. The translation takes place by multiplying from the left with the translator T and from the right with \tilde{T} (e.g., $Tx\tilde{T}$).

To express the rigid motion of an object we can apply a rotor and a translator simultaneously, this composition is equals to a motor $M = TR$. We apply the motor similarly as the rotor and the translator (e.g. $x = Mx\tilde{M}$). Surprisingly this formulation of the rigid motion can be applied not only to lines and points but also to all the entities of Table 3.1.

4 Omnidirectional Vision Using Conformal Geometric Algebra

The model defined by Geyer and Daniilidis [3] is used to find an equivalent spherical projection of a catadioptric projection. This model is very useful to simplify the projections, but the representation is not ideal because it is defined in a projective geometry context where the basis objects are points and lines and not the spheres. The computations are still complicated and difficult to follow.

Our proposal is based in the conformal geometric algebra where the basic element is the sphere. This is, all the entities (point, point pair, circle, plane) are defined in terms of the sphere (e.g., a point can be defined as sphere of zero radius). This framework also has the advantage that the intersection operation between entities is mathematically well defined (e.g. the intersection of a sphere with a line can be defined as $L \wedge S$). In short, the unified model is more natural and concise in the context of the conformal geometric algebra (of spheres) than the projective algebra (of points and lines). However the work of Geyer and Daniilidis [3] was extremely useful to accomplish the contribution of this paper.

4.1 Conformal Unified Model

We assume that the optical axis of the mirror is parallel to the e_2 axis, then let \mathbf{f} be a point in the Euclidean space (which represents the focus of the mirror which lies in such optical axis) defined by

$$\mathbf{f} = \alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3 \quad (31)$$

with conformal representation given by

$$F = \mathbf{f} + \frac{1}{2}\mathbf{f}^2 e + e_0 . \quad (32)$$

Using the point F as the center, we can define a unit sphere S (see Fig. 1) as follows

$$S = F - \frac{1}{2}e . \quad (33)$$

Now let N be the point of projection (that also lies on the optical axis) at a distance l of the point F , this point can be found using a translator

$$T = 1 + \frac{le_2 e}{2} \quad (34)$$

and then

$$N = T F \tilde{T} . \quad (35)$$

Finally, the image plane is perpendicular to the optical axis at a distance $-m$ from the point F and its equation is

$$\Pi = e_2 + (\mathbf{f} \cdot e_2 - m)e . \quad (36)$$

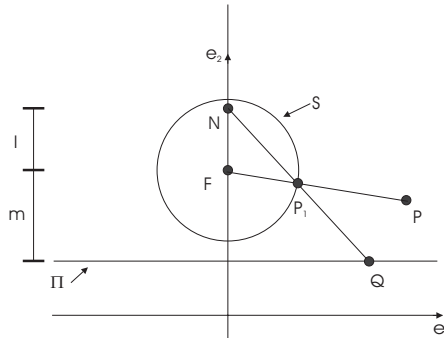


Fig. 1. Conformal Unify Model.

4.2 Point Projection

Let \mathbf{p} be a point in the Euclidean space, the corresponding homogeneous point in the conformal space is

$$P = \mathbf{p} + \frac{1}{2}\mathbf{p}^2 e + e_0 . \quad (37)$$

Now, for the projection of the point P we trace a line joining the points F and P , using the definition of the line in dual form we get

$$L_1^* = F \wedge P \wedge e . \quad (38)$$

Then, we calculate the intersections of the line L_1 and the sphere S which result in the point pair

$$PP^* = (L_1 \wedge S)^* . \quad (39)$$

From the point pair we choose the point P_1 which is the closest point to P , and then we find the line passing through the points P_1 and N

$$L_2^* = P_1 \wedge N \wedge e \quad (40)$$

Finally we find the intersection of the line L_2 with the plane Π

$$Q = (L_2 \wedge \Pi)^* . \quad (41)$$

The point Q is the projection in the image plane of the point P of the space. Notice that we can project any point in the space into any type of mirror (changing l and m) using the previous procedure (see Fig. 2). The reader can now see how simple and elegant is the treatment of the unified model in the conformal geometric algebra.

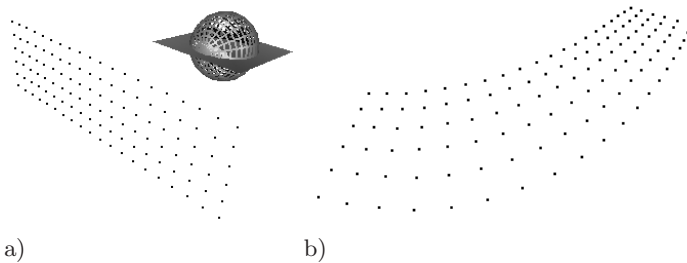


Fig. 2. a) Unified Model and points in the space. b) Projection in the image plane.

4.3 Inverse Point Projection

We have already seen how to project a point in the space to the image plane through the sphere. But now we want to back-project a point in the image plane

into 3D space. First, let Q be a point in such image plane, the equation of the line passing through the points Q and N is

$$L_2^* = Q \wedge N \wedge e, \quad (42)$$

and the intersection of the line L_2 and the sphere S is

$$PP^* = (L_2 \wedge S)^*. \quad (43)$$

From the point pair we choose the point P_1 which is the closest point to Q , and then we find the equation of the line from the point P_1 and the focus F

$$L_1^* = P_1 \wedge F \wedge e. \quad (44)$$

The point P lies on the line L_1^* , but it can not be calculated exactly because a coordinate has been lost when the point was projected to the image plane (a single view does not allow to know the projective depth). However, we can project this point to some plane and say that it is equivalent to the original point up to a scale factor (see Fig. 3).

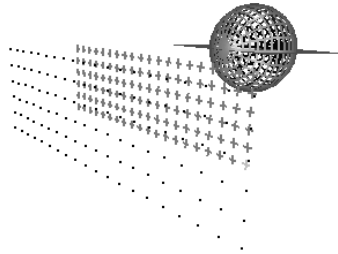


Fig. 3. Inverse point projection (from the image to the space). The crosses are the projected points and the dots are the original points.

4.4 Line Projection

Suppose that L is a line in the space and we want to find its projection in the image plane (see Fig. 4.a). First we find the plane were L and F lies, its equation is

$$P^* = L^* \wedge F. \quad (45)$$

The intersection of the plane and the sphere is the great circle defined as

$$C^* = (P \wedge S). \quad (46)$$

The line that passes through the center of the circle and is perpendicular to the plane P is

$$U^* = (C \wedge e). \quad (47)$$

Using U as an axis we make a rotor

$$R = \exp\left(\frac{\theta}{2}U\right). \quad (48)$$

We find a point pair PP^* that lies on the circle with

$$PP^* = (C \wedge e_2)^*. \quad (49)$$

We choose any point from the point pair, say P_1 , and using the rotor R we can find the points in the circle

$$P'_1 = RP_1\tilde{R}, \quad (50)$$

for each point P'_1 we find the line that passes through the points P_1 and N defined as

$$L_2^* = P'_1 \wedge N \wedge e. \quad (51)$$

Finally for each line L_2 we find the intersection with the plane Π

$$P_2 = (L_2 \wedge \Pi)^*, \quad (52)$$

which is the projection of the line in the space to the image plane (see Fig. 4.b).

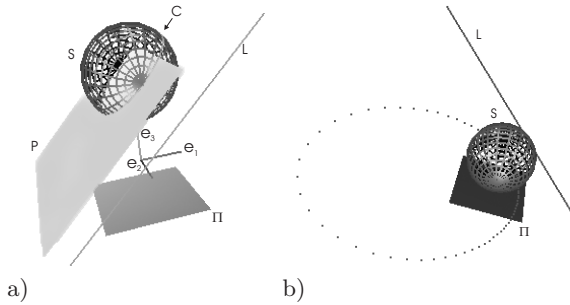


Fig. 4. a) Line projection to the sphere. b) Line projection to the image plane (note that in this case results an ellipse).

5 Experimental Analysis

In this experiment a robot was placed in the corridor and the goal was that the robot passes through the corridor using purely the information extracted from the omnidirectional image of a calibrated parabolic system (see Fig. 5). As we know, parabolic mirror projects lines in the space into circles in the image and due to the conformal projection angles are preserved. In this experiment we take advantage of these properties in order to control the robot by means of circles in the omnidirectional image. With the axes of the circles we can calculate the

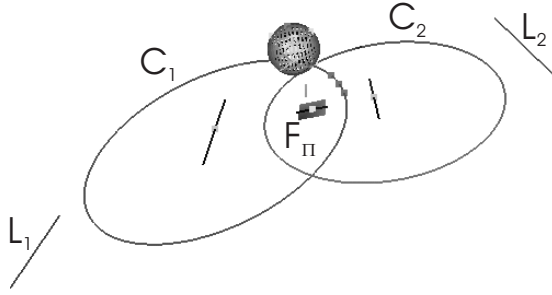


Fig. 5. Robot navigation control using circles in the image plane (dark lines represent the axes of the circles).

robot heading. If we want the robot in the center of the corridor the radius of the circles must be the same, and then the distances to the corridor lines will also be the same. Note that Π is the image plane (36) and F_{Π} is the focus F of the mirror projected in the plane Π .

The next computations are illustrated in figure 6. The first circle C_1^* (dual representation according to Table 3.1) is defined with the wedge of three points

$$C_1^* = X_1 \wedge X_2 \wedge X_3 , \quad (53)$$

similarly for the second circle C_2^*

$$C_2^* = X_4 \wedge X_5 \wedge X_6 . \quad (54)$$

The center of the each circle ($i=1,2$) is calculated by

$$N_i^* = (C_i \wedge e) \cdot \Pi . \quad (55)$$

The first axis of each circle is defined as

$$A_{1,i}^* = N_i \wedge F_{\Pi} \wedge e , \quad (56)$$

an the second axis is

$$A_{2,i}^* = \{ [((A_{1,i} \wedge e) \cdot e_0) \cdot I_e] \cdot (e_{12}) \} \wedge N_i \wedge e . \quad (57)$$

The angle in the image is

$$\theta = \arccos(A_{1,1} \cdot e_{3+-}) \quad (58)$$

The sphere with center and radius of the circle is

$$S_i = C_i / \Pi . \quad (59)$$

and the radius of the circle is

$$r_i = S_i \cdot S_i . \quad (60)$$

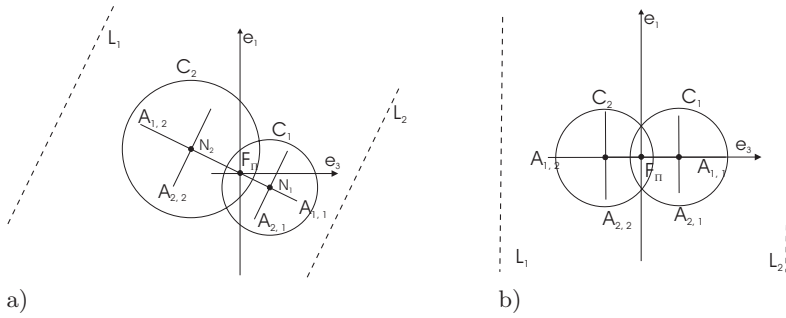


Fig. 6. a) In this image the robot is not parallel to the corridor nor centered. b) In this image the robot is parallel to the corridor and centered.

The control strategy for the navigation is based (as we said previously) on the angles of the circles and its radius. With the angles we correct the parallelism of the robot with the corridor. Furthermore, we place the robot at the center of the corridor using the radius of the circles. The position error is defined by

$$\alpha = (r_1 - r_2)\pi/2. \quad (61)$$

The heading error angle is calculated as

$$\beta = \theta - \pi/2. \quad (62)$$

The robot angular velocity is calculated with the combination of the robot heading error, the position error and the proportional gain κ

$$\omega = \kappa(\beta + \alpha). \quad (63)$$

6 Conclusions

The major contribution of this paper is the refinement and improvement of the use of the unified model for omnidirectional vision. To achieve this goal the authors used the conformal geometric algebra, a modern framework for the projective space of hyper-spheres. This framework is equipped with homogeneous representations of points, lines, planes and spheres, operations of incidence algebra and conformal transformations expressed effectively as versors. The authors show how the analysis of diverse catadioptric mirrors becomes transparent and computationally simpler. As a result, the algebraic burden is reduced for the users who can now develop more efficient algorithms for omnidirectional vision. The paper includes complementary experimental analysis of omnidirectional vision guided robot navigation.

Acknowledgment. We are thankful to CONACYT and CINVESTAV for supporting this work.

References

1. Baker, S., Nayar, S.: A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision* (1999) 35(2):1-22.
2. Gaspar, J., Santos-Victor, J.: Visual path following with a catadioptric panoramic camera. In *Proceedings of the 7th International Symposium on Intelligent Robotic Systems (SIRS'99)*, Coimbra, Portugal (1999) 139-147.
3. Geyer, C., Daniilidis, K.: Catadioptric projective geometry. *International Journal of Computer Vision* (2001) 223-243.
4. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, UK (2001).
5. Hestenes, D., Li, H., Rockwood, A.: New algebraic tools for classical geometry. In Sommer, G.(ed.): *Geometric Computing with Clifford Algebras*. Springer-Verlag, Berlin Heidelberg (2001) 3-23.
6. Winters, N., Santos-Victor, J.: Omni-directional visual navigation. In *Proceedings of the 7th International Symposium on Intelligent Robotic Systems (SIRS'99)*, Coimbra, Portugal (1999) 109-118.

A Robust Algorithm for Characterizing Anisotropic Local Structures

Kazunori Okada¹, Dorin Comaniciu¹, Navneet Dalal², and Arun Krishnan³

¹ Real-Time Vision & Modeling Department
Siemens Corporate Research, Inc.

755 College Road East, Princeton, NJ 08540, USA

² INRIA Rhône-Alpes

655, avenue de l'Europe 38330 Montbonnot, France

³ CAD Program

Siemens Medical Solutions USA, Inc.

51 Valley Stream Parkway, Malvern, PA 19355, USA

Abstract. This paper proposes a robust estimation and validation framework for characterizing local structures in a positive multi-variate continuous function approximated by a Gaussian-based model. The new solution is robust against data with large deviations from the model and margin-truncations induced by neighboring structures. To this goal, it unifies robust statistical estimation for parametric model fitting and multi-scale analysis based on continuous scale-space theory. The unification is realized by formally extending the mean shift-based density analysis towards continuous signals whose local structure is characterized by an anisotropic fully-parameterized covariance matrix. A statistical validation method based on analyzing residual error of the chi-square fitting is also proposed to complement this estimation framework. The strength of our solution is the aforementioned robustness. Experiments with synthetic 1D and 2D data clearly demonstrate this advantage in comparison with the γ -normalized Laplacian approach [12] and the standard sample estimation approach [13, p.179]. The new framework is applied to 3D volumetric analysis of lung tumors. A 3D implementation is evaluated with high-resolution CT images of 14 patients with 77 tumors, including 6 part-solid or ground-glass opacity nodules that are highly non-Gaussian and clinically significant. Our system accurately estimated 3D anisotropic spread and orientation for 82% of the total tumors and also correctly rejected all the failures without any false rejection and false acceptance. This system processes each 32-voxel volume-of-interest by an average of two seconds with a 2.4GHz Intel CPU. Our framework is generic and can be applied for the analysis of blob-like structures in various other applications.

1 Introduction

This paper presents a robust estimation and validation framework for characterizing a d -variate positive function that can be locally approximated by a

Gaussian-based model. Gaussian model fitting is a well-studied standard technique [4, ch.2]. However, it is not trivial to fit such a model to data with outliers and margin-truncation induced by neighboring structures. For example, minimum volume ellipsoid covariance estimator [17] addresses the robustness to the outliers however its effectiveness is limited regarding the truncation issue. Fig.1 illustrates our problem with some real medical imaging examples of lung tumors in 3D CT data. The figure shows 2D dissections and 1D profiles of two tumors. The symbol \mathbf{x} and solid-line ellipses denote our method's estimates. In developing an algorithm to describe the tumors, our solution must be robust against 1) influences from surrounding structures (i.e., margin-truncation: Fig.1a,b), 2) deviation of the signal from a Gaussian model (i.e., non-Gaussianity: Fig.1c,d), and 3) uncertainty in the given marker location (i.e., initialization: Fig.1a,c).

Our proposed solution unifies robust statistical methods for density gradient estimation [3] and continuous linear scale-space theory [21,9,12]. By likening the arbitrary positive function describing an image signal to the probability density function, the mean shift-based analysis is further developed towards 1) Gaussian model fitting to a continuous positive function and 2) anisotropic fully-parameterized covariance estimation. Its robustness is due to the multi-scale nature of this framework that implicitly exploits the scale-space function. A statistical validation method based on chi-square analysis is also proposed to complement this robust estimation framework. Sections 2 and 3 formally describe our solution. The robustness is empirically studied with synthetic data and the results are described in Section 4.1.

1.1 Medical Imaging Applications

One of the key problems in the volumetric medical image analysis is to characterize the 3D local structure of tumors across various scales. The size and shape of tumors vary largely in practice. Such underlining scales of tumors also provide important clinical information, correlating highly with probability of malignancy. A large number of studies have been accumulated for automatic detection and characterization of lung nodules [19]. Several recent studies (e.g., [10,18]) exploited 3D information of nodules provided in X-ray computed-tomography (CT) images. However, these methods, based on the template matching technique, assumed the nodules to be spherical. Recent clinical studies suggested that part- and non-solid or ground-glass opacity (GGO) nodules, whose shape deviates largely from such a spherical model (Fig.1c,d), are more likely to be malignant than solid ones [6]. One of our motivations of this study is to address this clinical demand by considering the robust estimation of 3D tumor spread and orientation with non-spherical modeling. We evaluate the proposed framework applied for the pulmonary CT data in Section 4.2.

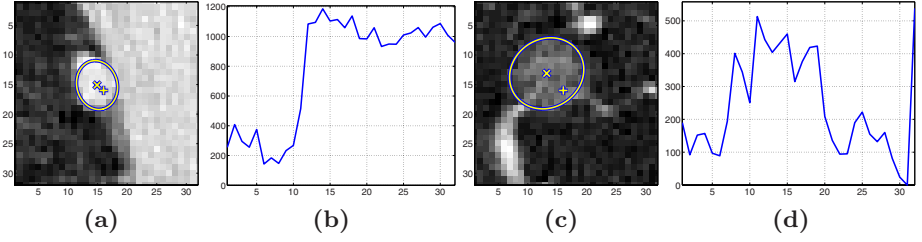


Fig. 1. An illustration of our problem with lung tumor examples captured in 3D CT data. From left to right, (a): on-the-wall tumor in 2D dissection, (b): 1D horizontal profile of (a) through the tumor center, (c): non-solid (GGO) tumor, and (d): 1D vertical profile of (c). “+” denotes markers used as initialization points provided by expert radiologists. Our method’s estimates of the tumor center and anisotropic spread are shown by “x” and 50% confidence ellipses, respectively.

2 Multi-scale Analysis of Local Structure

2.1 Signal Model

Given a d -dimensional continuous signal $f(\mathbf{x})$ with non-negative values, we use the symbol \mathbf{u} for describing the location of a spatial local maximum of f (or a mode in the sense of density estimation). Suppose that the local region of f around \mathbf{u} can be approximated by a product of a d -variate Gaussian function and a positive multiplicative parameter,

$$f(\mathbf{x}) \simeq \alpha \times \Phi(\mathbf{x}; \mathbf{u}, \Sigma) |_{\mathbf{x} \in \mathcal{S}} \quad (1)$$

$$\Phi(\mathbf{x}; \mathbf{u}, \Sigma) = (2\pi)^{-d/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{u})^t \Sigma^{-1} (\mathbf{x} - \mathbf{u})\right) \quad (2)$$

where \mathcal{S} is a set of data points in the neighborhood of \mathbf{u} , belonging to the basin of attraction of \mathbf{u} . An alternative is to consider a model with a DC component $\beta \geq 0$ so that $f \simeq \alpha \times \Phi + \beta$. It is, however, straightforward to locally offset the DC component. Thus we will not consider it within our estimation framework favoring a simpler form. Later, we will revisit this extended model for the statistical validation of the resulting estimates. The problem of our interest can now be understood as the parametric model fitting and the estimation of the model parameters: mean \mathbf{u} , covariance Σ , and amplitude α . The *mean* and *covariance* of Φ describe the *spatial local maximum* and *spread* of the local structure, respectively. The anisotropy of such structure can be specified only by a fully-parameterized covariance.

2.2 Scale-Space Representation

The scale-space theory [21,9,12] states that, given any d -dimensional continuous signal $f : \mathcal{R}^d \rightarrow \mathcal{R}$, the scale-space representation $F : \mathcal{R}^d \times \mathcal{R}_+ \rightarrow \mathcal{R}$ of f is

defined to be the solution of the diffusion equation, $\partial_h F = 1/2\nabla^2 F$, or equivalently the convolution of the signal with Gaussian kernels $\Phi(\mathbf{x}; \mathbf{0}, \mathbf{H})$ of various bandwidths (or scales) $\mathbf{H} \in \mathcal{R}^{d \times d}$,

$$F(\mathbf{x}; \mathbf{H}) = f(\mathbf{x}) * \Phi(\mathbf{x}; \mathbf{0}, \mathbf{H}). \quad (3)$$

When $\mathbf{H} = h\mathbf{I}$ ($h > 0$), F represents the solution of the isotropic diffusion process [12] and also the Tikhonov regularized solution of a functional minimization problem, assuming that scale invariance and semi-group constraints are satisfied [14]. When \mathbf{H} is allowed to be a fully parameterized symmetric positive definite matrix, F represents the solution of an anisotropic *homogeneous* diffusion process $\partial_{\mathbf{H}} F = 1/2\nabla\nabla^t F$ that is related, but not equivalent, to the well-known anisotropic diffusion [15].

2.3 Mean Shift Procedure for Continuous Scale-Space Signal

In this section, we further develop the fixed-bandwidth mean shift [2], introduced previously for the non-parametric point density estimation, towards the analysis of continuous signal evaluated in the linear scale-space.

The gradient of the scale-space representation $F(\mathbf{x}; \mathbf{H})$ can be written as convolution of f with the DOG kernel $\nabla\Phi$, since the gradient operator commutes across the convolution operation. Some algebra reveals that ∇F can be expressed as a function of a vector whose form resembles the density mean shift,

$$\begin{aligned} \nabla F(\mathbf{x}; \mathbf{H}) &= f(\mathbf{x}) * \nabla\Phi(\mathbf{x}; \mathbf{H}) \\ &= \int f(\mathbf{x}')\Phi(\mathbf{x} - \mathbf{x}'; \mathbf{H})\mathbf{H}^{-1}(\mathbf{x}' - \mathbf{x})d\mathbf{x}' \\ &= \mathbf{H}^{-1}F(\mathbf{x}; \mathbf{H})\mathbf{m}(\mathbf{x}; \mathbf{H}) \end{aligned} \quad (4)$$

$$\mathbf{m}(\mathbf{x}; \mathbf{H}) \equiv \frac{\int \mathbf{x}'\Phi(\mathbf{x} - \mathbf{x}'; \mathbf{H})f(\mathbf{x}')d\mathbf{x}'}{\int \Phi(\mathbf{x} - \mathbf{x}'; \mathbf{H})f(\mathbf{x}')d\mathbf{x}'} - \mathbf{x}. \quad (5)$$

Eq.(5) defines the extended fixed-bandwidth mean shift vector for f . Setting $f(\mathbf{x}') = 1$ in Eq.(5) results in the same form as the density mean shift vector. Note however that \mathbf{x} in Eq.(5) is an ordinal variable while a random variable was considered in [2]. Eq.(5) can be seen as introducing a weight variable $w \equiv f(\mathbf{x}')$ to the kernel $\Phi(\mathbf{x} - \mathbf{x}')$. Therefore, an arithmetic mean of \mathbf{x}' in our case is not weighted by the Gaussian kernel but by its product with the signal $\Phi(\mathbf{x} - \mathbf{x}')f(\mathbf{x}')$.

The mean shift procedure [3] is defined as iterative updates of a data point \mathbf{x}_i until its convergence at \mathbf{y}_i^m ,

$$\mathbf{y}_{j+1} = \mathbf{m}(\mathbf{y}_j; \mathbf{H}) + \mathbf{y}_j; \quad \mathbf{y}_0 = \mathbf{x}_i. \quad (6)$$

Such iteration gives a robust and efficient algorithm of gradient-ascent, since $\mathbf{m}(\mathbf{x}; \mathbf{H})$ can be interpreted as a normalized gradient by rewriting Eq.(4); $\mathbf{m}(\mathbf{x}; \mathbf{H}) = \mathbf{H}\nabla F(\mathbf{x}; \mathbf{H})/F(\mathbf{x}; \mathbf{H})$. F is strictly non-negative valued since f is assumed to be non-negative. Therefore, the direction of the mean shift vector aligns with the exact gradient direction when \mathbf{H} is isotropic with a positive scale.

2.4 Finding Spatial Local Maxima

We assume that the signal is given with information of where the target structure is roughly located but we do not have explicit knowledge of its spread. The marker point \mathbf{x}_p indicates such location information. We allow \mathbf{x}_p to be placed anywhere within the basin of attraction \mathcal{S} of the target structure. To increase the robustness of this approach, we run N_1 mean shift procedures initialized by sampling the neighborhood of \mathbf{x}_p uniformly. The majority of the procedure's convergence at the same location indicates the location of the maximum. The point proximity is defined by using the Mahalanobis distance with \mathbf{H} . This approach is efficient because it does not require the time-consuming explicit construction of $F(\mathbf{x}; \mathbf{H})$ from $f(\mathbf{x})$.

2.5 Robust Anisotropic Covariance Estimation by Constrained Least-Squares in the Basin of Attraction

In the sequel we estimate the fully parameterized covariance matrix Σ in Eq.(1), characterizing the d -dimensional anisotropic spread and orientation of the signal f around the local maximum \mathbf{u} . Classical scale-space approaches relying on the γ -normalized Laplacian [12] are limited to the isotropic case thus not applicable to this problem. Another approach is the standard sample estimation of Σ by treating f as a density function [13, p.179]. However, this approach becomes suboptimal in the presence of the margin-truncations. Addressing this issue, we present a constrained least-squares framework for the estimation of the anisotropic fully-parameterized covariance of interest based on the mean shift vectors collected in the basin of attraction of \mathbf{u} .

With the signal model of Eq.(1), the definition of the mean shift vector of Eq.(5) can be rewritten as a function of Σ ,

$$\begin{aligned} \mathbf{m}(\mathbf{y}_j; \mathbf{H}) &= \mathbf{H} \frac{\nabla F(\mathbf{y}_j; \mathbf{H})}{F(\mathbf{y}_j; \mathbf{H})} \\ &\simeq \mathbf{H} \frac{\alpha \Phi(\mathbf{y}_j; \mathbf{u}, \Sigma + \mathbf{H})(\Sigma + \mathbf{H})^{-1}(\mathbf{u} - \mathbf{y}_j)}{\alpha \Phi(\mathbf{y}_j; \mathbf{u}, \Sigma + \mathbf{H})} \\ &= \mathbf{H}(\Sigma + \mathbf{H})^{-1}(\mathbf{u} - \mathbf{y}_j). \end{aligned} \quad (7)$$

Further rewriting Eq.(7) results in a linear matrix equation of unknown Σ ,

$$\Sigma \mathbf{H}^{-1} \mathbf{m}_j = \mathbf{b}_j \quad (8)$$

where $\mathbf{m}_j \equiv \mathbf{m}(\mathbf{y}_j; \mathbf{H})$ and $\mathbf{b}_j \equiv \mathbf{u} - \mathbf{y}_j - \mathbf{m}_j$.

An over-complete set of the linear equations can be formed by using all the trajectory points $\{\mathbf{y}_j | j = 1, \dots, t_u\}$ located within the basin of attraction \mathcal{S} . For efficiently collecting a sufficient number of samples $\{(\mathbf{y}_j, \mathbf{m}_j)\}$, we run N_2 mean shift procedures initialized by sampling the neighborhood of \mathbf{u} uniformly. This results in t_u samples ($t_u = \sum_{i=1}^{N_2} t_i$), where t_i denotes the number of points

on the trajectory starting from x_i . The system described in Eq.(8) is solved by considering the following constrained least-squares problem [7,5],

$$\begin{aligned}\mathbf{A}\boldsymbol{\Sigma} &= \mathbf{B} \\ \boldsymbol{\Sigma} &\in \mathcal{SPD} \\ \mathbf{A} &= (\mathbf{m}_1, \dots, \mathbf{m}_{t_u})^t \mathbf{H}^{-t} \\ \mathbf{B} &= (\mathbf{b}_1, \dots, \mathbf{b}_{t_u})^t\end{aligned}\tag{9}$$

where \mathcal{SPD} denotes a set of symmetric positive definite matrices in $\mathcal{R}^{d \times d}$.

Following [1], the unique solution $\boldsymbol{\Sigma}^*$ of Eq.(9) is expressed by,

$$\boldsymbol{\Sigma}^* = \mathbf{U}_P \boldsymbol{\Sigma}_P^{-1} \mathbf{U}_{\tilde{\mathbf{Q}}} \boldsymbol{\Sigma}_{\tilde{\mathbf{Q}}} \mathbf{U}_{\tilde{\mathbf{Q}}}^t \boldsymbol{\Sigma}_P^{-1} \mathbf{U}_P^t \tag{10}$$

which involves symmetric Schur decompositions [5, p.393] of the matrices $\mathbf{P} \equiv \mathbf{A}^t \mathbf{A}$ and $\tilde{\mathbf{Q}} \equiv \boldsymbol{\Sigma}_P \mathbf{U}_P^t \mathbf{Q} \mathbf{U}_P \boldsymbol{\Sigma}_P$ given $\mathbf{Q} \equiv \mathbf{B}^t \mathbf{B}$, i.e.,

$$\begin{aligned}\mathbf{P} &= \mathbf{U}_P \boldsymbol{\Sigma}_P^2 \mathbf{U}_P^t \\ \tilde{\mathbf{Q}} &= \mathbf{U}_{\tilde{\mathbf{Q}}} \boldsymbol{\Sigma}_{\tilde{\mathbf{Q}}}^2 \mathbf{U}_{\tilde{\mathbf{Q}}}^t.\end{aligned}$$

The solution $\boldsymbol{\Sigma}^*$ is derived from finding \mathbf{Y}^* in the Cholesky factorization of $\boldsymbol{\Sigma} = \mathbf{Y} \mathbf{Y}^t$. It can be shown that $\boldsymbol{\Sigma}^*$ uniquely minimizes an area criterion $\|\mathbf{A} \mathbf{Y} - \mathbf{B} \mathbf{Y}^{-t}\|_F^2$ where $\|\cdot\|_F$ denotes the Frobenius norm. This area criterion is related to the total least-squares [20] since errors in both \mathbf{A} and \mathbf{B} are considered for the minimization.

2.6 Scale Selection Criterion

The multi-scale analysis treats \mathbf{H} as a variable parameter. It is supposed that a set of analysis bandwidths $\{\mathbf{H}_k | k = 1, \dots, K\}$ is given *a priori*. Our scale selection criterion is based on the stability test [2]. Given a set of estimates $\{(\mathbf{u}_k, \boldsymbol{\Sigma}_k)\}$ for a series of the successive analysis bandwidths, a form of the Jensen-Shannon divergence is defined by,

$$JS(k) = \frac{1}{2} \log \frac{\frac{1}{2a+1} \sum_{i=k-a}^{k+a} |\boldsymbol{\Sigma}_i|}{\sqrt[2a+1]{\prod_{i=k-a}^{k+a} |\boldsymbol{\Sigma}_i|}} + \frac{1}{2} \sum_{i=k-a}^{k+a} (\mathbf{u}_i - \mathbf{u})^t \left(\sum_{i=k-a}^{k+a} \boldsymbol{\Sigma}_i \right)^{-1} (\mathbf{u}_i - \mathbf{u}) \tag{11}$$

where $\mathbf{u} = \frac{1}{2a+1} \sum_{i=k-a}^{k+a} \mathbf{u}_i$ and a define the neighborhood width of the divergence computation. The most stable estimate across the analysis bandwidths provides a local minimum of the divergence profile. We treat this result as the final estimation of our multi-scale analysis.

3 Statistical Validation

In this section, we present a goodness-of-fit measure for validating the resulting estimates. Such statistical validation gives a principled means for rejecting

accidental ill-estimates. We treat this problem as analysis of chi-square fitting residual errors. We employ a linear model with an additive parameter of the DC component; $f \simeq \alpha \times \Phi + \beta$. Recall that our estimation model is without the DC. The additional degree of freedom introduced serves as another goodness-of-fit indicator. Given the estimate pair (\mathbf{u}^*, Σ^*) , the following defines the signal response estimate \hat{f} with two unknowns,

$$\hat{f}(\mathbf{x}, \mathbf{u}^*, \Sigma^*; \alpha, \beta) = \alpha \times \Phi(\mathbf{x}; \mathbf{u}^*, \Sigma^*) + \beta|_{\mathbf{x} \in \mathcal{S}}. \quad (12)$$

The chi-square statistic indicates the residual error of the fitted model $\hat{f}(\mathbf{x})$ [16, p.660],

$$\chi^2 \equiv \sum_{i \in \mathcal{S}} \left(\frac{f(\mathbf{x}_i) - \hat{f}(\mathbf{x}_i)}{\sigma_i} \right)^2 = \sum_{i \in \mathcal{S}} \left(\frac{f(\mathbf{x}_i) - \alpha \Phi(\mathbf{x}_i) - \beta}{\sigma_i} \right)^2 \quad (13)$$

where σ_i is local uncertainty of normally distributed error $(f(\mathbf{x}_i) - \hat{f}(\mathbf{x}_i))^2$.

Parameters α and β are estimated by chi-square fitting. Since both are non-negative, we introduce parameters a and b such that $\alpha = a^2$ and $\beta = b^2$. The estimates α^* and β^* are given by solving $\partial \chi^2 / \partial a = 0$ and $\partial \chi^2 / \partial b = 0$,

$$(\alpha^*, \beta^*) = \begin{cases} (p, q) & \text{if } p > 0 \text{ and } q > 0 \\ \left(\frac{\sum f(\mathbf{x}_i) \Phi(\mathbf{x}_i)}{\sum \Phi(\mathbf{x}_i)^2}, 0 \right) & \text{if } p > 0 \text{ and } q \leq 0 \\ \left(0, \frac{\sum f(\mathbf{x}_i)}{N_s} \right) & \text{if } p \leq 0 \text{ and } q > 0 \\ (0, 0) & \text{if } p \leq 0 \text{ and } q \leq 0 \end{cases} \quad (14)$$

where $\sigma = \sigma_i$ for all i ,

$$p = \frac{N_s \sum f(\mathbf{x}_i) \Phi(\mathbf{x}_i) - \sum f(\mathbf{x}_i) \sum \Phi(\mathbf{x}_i)}{N_s \sum \Phi(\mathbf{x}_i)^2 - (\sum \Phi(\mathbf{x}_i))^2} \quad (15)$$

$$q = \frac{\sum f(\mathbf{x}_i) \sum \Phi(\mathbf{x}_i)^2 - \sum \Phi(\mathbf{x}_i) \sum f(\mathbf{x}_i) \Phi(\mathbf{x}_i)}{N_s \sum \Phi(\mathbf{x}_i)^2 - (\sum \Phi(\mathbf{x}_i))^2} \quad (16)$$

and N_s is the number of samples in \mathcal{S} and all the summations are over $i \in \mathcal{S}$.

Given the above parameter estimates, χ^2 is computed by using Eq.(13). Chi-square probability function Q [16, p.221] is employed to indicate an ill-fit of our model to the given signal,

$$Q(\chi^2 | \nu) = Q\left(\frac{N_s - M}{2}, \frac{\chi^2}{2}\right) = g\left(\frac{N_s - M}{2}, \frac{\chi^2}{2}\right). \quad (17)$$

In Eq.(17), g is the incomplete gamma function [16, ch.6.2] with the number of degrees of freedom $\nu = (N - M)/2$, and M is the number of parameters.

Finally, we obtain the following rejection criterion,

$$\text{Reject } (\mathbf{u}^*, \Sigma^*) \text{ if } Q < th_1 \text{ or } \beta^* > th_2. \quad (18)$$

The threshold for Q is set conservatively to the common confidence level $th_1 = 0.001$ [16, p.664]. Having a large estimate for β also indicates an ill-fit with our estimation model without the DC. The threshold th_2 for β can be learned from training samples for specific applications.

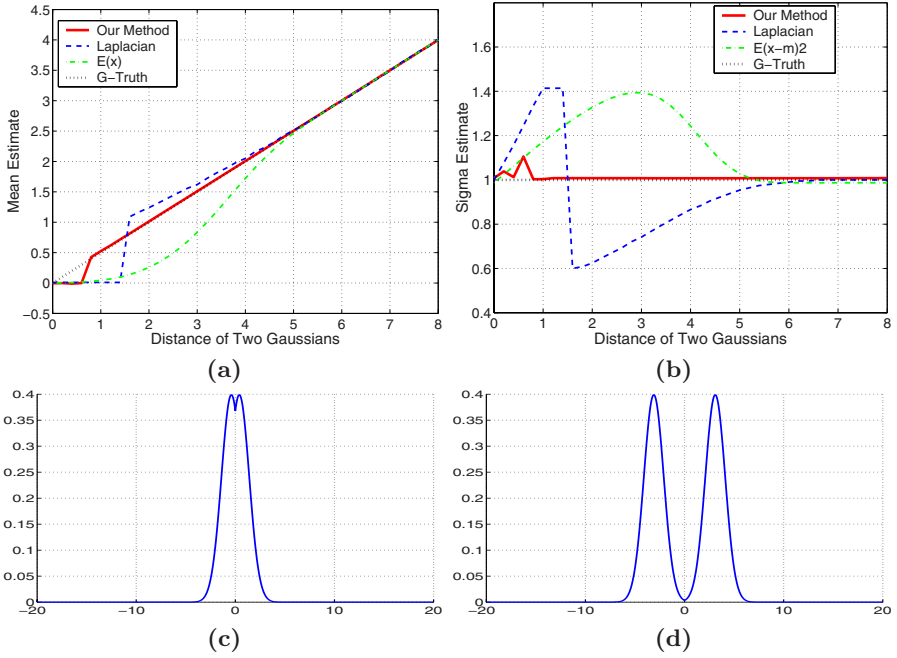


Fig. 2. Comparison of our method (solid-line) with γ -normalized Laplacian (dashed-line) and standard sample estimate (dot-dashed-line) using 1D synthetic data. The ground-truth $u = D/2$ and $\sigma = 1$ are denoted by dotted-line. Test data is generated by superimposing two Gaussians with a varying distance D for evaluating robustness of estimates against biases caused by neighboring structures. (a): local maxima estimates, (b): scale estimates, (c): our method’s break-point $D = 0.8$, below which estimations are subjected to the bias. (c): γ -normalized Laplacian’s break-point $D = 6.2$.

4 Experiments

4.1 Synthetic Data

The proposed framework is examined with 1D and 2D synthetic data. Fig.2 compares local maximum and scale estimates by a 1D implementation of our algorithm with those by the γ -normalized Laplacian [12] and the standard sample estimation [13, p.179]. The test data is generated at each location by taking the maximum of two superimposed 1D Gaussians offset by a varying distance D . Each Gaussian has the same variance $\sigma = 1$ and height $\alpha = 1$. The 1D system employs all the available data points ($N_1 = N_2 = N_S$) and 40 analysis scales with 0.05 interval ($h = (0.1^2, 0.15^2, \dots, 2^2)$ for $\mathbf{H} = h\mathbf{I}$). For the sample variance estimation, the densities $p(x_i)$ are approximated by $f(x_i)$ normalized by the probability mass within $\pm 1\sigma$ around the true maximum. The results indicated that our method achieved robust and accurate estimations even with the presence of the severe margin-truncations, clearly demonstrating the advantage of our framework. Fig.3 shows examples with 2D synthetic data. Estimates, shown as 50%

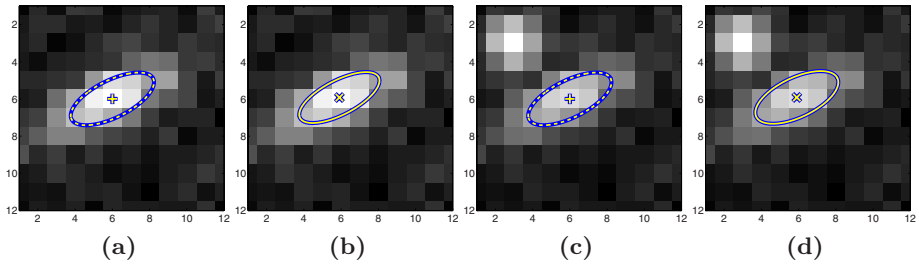


Fig. 3. Examples with 2D synthetic data. (a) and (b) illustrate the ground-truth and our method’s estimate for an anisotropic Gaussian $\Sigma=[2 \ -2; -2 \ 5]$ with random additive noise. (c) and (d) show those for two Gaussians with the noise. The center of the smaller Gaussian is deviated by 4 Mahalanobis distance away from the target Gaussian. “+” and dashed-ellipses indicate ground-truth local maximum and spread. “x” and solid-ellipses display those estimated by our 2D algorithm.

confidence ellipses, by a 2D implementation of our method are compared for two types of test data in the presence of random noise. This 2D implementation utilizes all available data points and 12 analysis scales ($h = (0.5^2, 0.75^2, \dots, 3.25^2)$). The results are almost identical to the ground-truth despite the presence of the random noise and the neighboring structure.

4.2 Lung HRCT Data

A 3D implementation of the proposed algorithm is evaluated with high-resolution computed-tomography (HRCT) images of pulmonary tumors. Each volumetric image consists of 12-bit positive values over an array of 512x512 lattices.

A straightforward implementation of our algorithm without any 3D specific adaptation provides the 3D tumor analysis system. A set of analysis bandwidths (18 scales with 0.25 interval $h = (0.50^2, 0.75^2, \dots, 4.75^2)$) and markers indicating rough tumor locations are given to the system *a priori*. The marker locations are provided by expert radiologists, however most of the markers deviate from the tumor centers with a certain degree. We use uniform sampling in the 3-voxel neighborhood of the marker (i.e., $N_1 = 7$). The same strategy is employed for initializing the mean shift trajectories around the local maximum (i.e., $N_2 = 7$). The neighborhood width of the divergence computation is set to $a = 1$ (considering only three adjacent scales). For the validation, all data points that lie within the 90% confidence ellipsoid of (\mathbf{u}^*, Σ^*) are used. The degrees of freedom in Eq.(17) are given by $M = 3 + 6 + 2 = 11$. The β threshold in Criterion(18) is set to $th_2 = 400$. The global uncertainty σ in Eq.(13) is estimated from the sample variance of 77 tumor data, resulting in $\sigma = 356$. This tumor analysis system is implemented in C language and processes each 32-voxel volume-of-interest (VOI) by an average of two seconds with a 2.4GHz Intel CPU.

HRCT data of 14 patients displaying the total of 77 pulmonary tumors were used for this evaluation. 63 cases resulted in successful estimation confirmed by expert inspection. All the solitary tumors were correctly estimated. Most of the

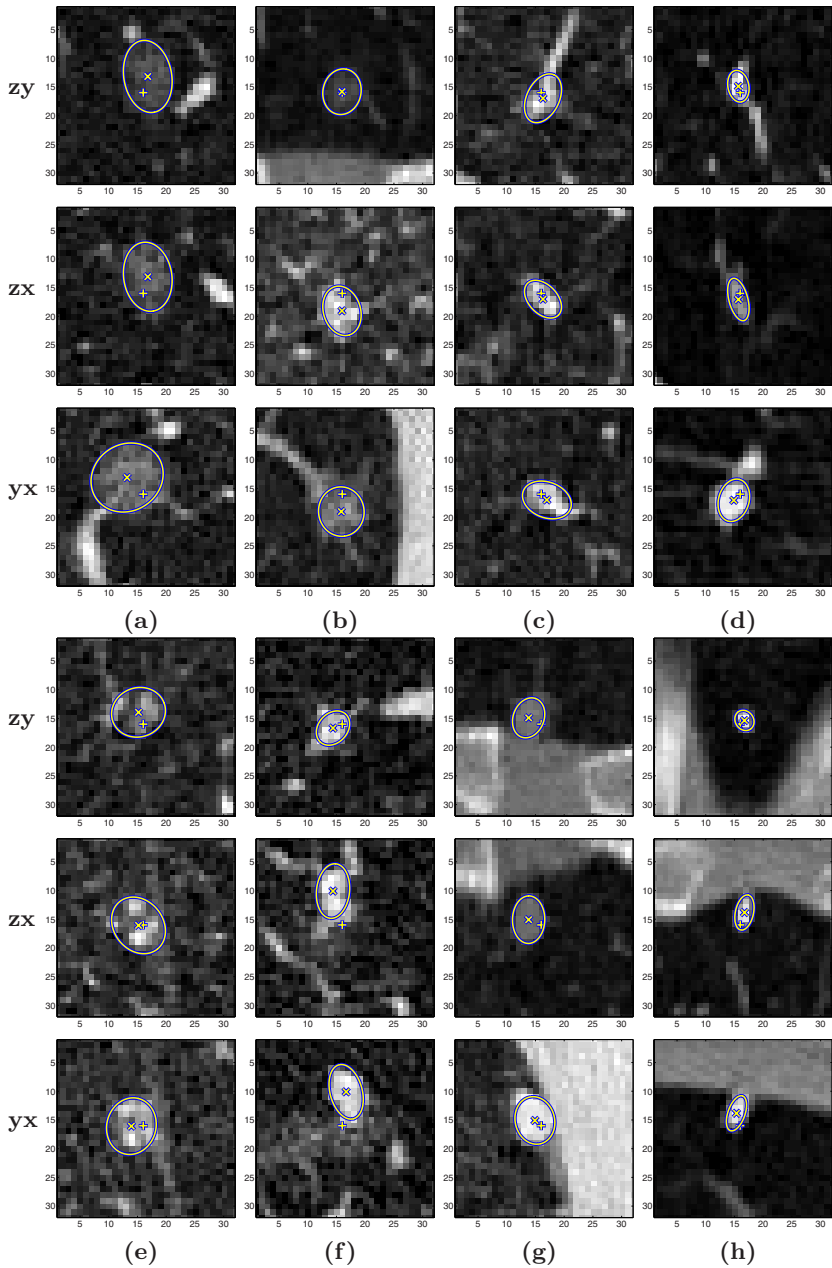


Fig. 4. Examples of the estimation results with 3D HRCT data. The marker locations are indicated by “+”. The estimated local maxima are indicated by “x”. The estimated spread of the tumors are shown as 2D intersections of 50% confidence ellipsoids. Cases (a) and (b) are GGO nodules identified by experts. Cases (c) to (f) are tumors with irregular non-spherical shapes. Cases (g) and (h) illustrate tumors on the lung wall.

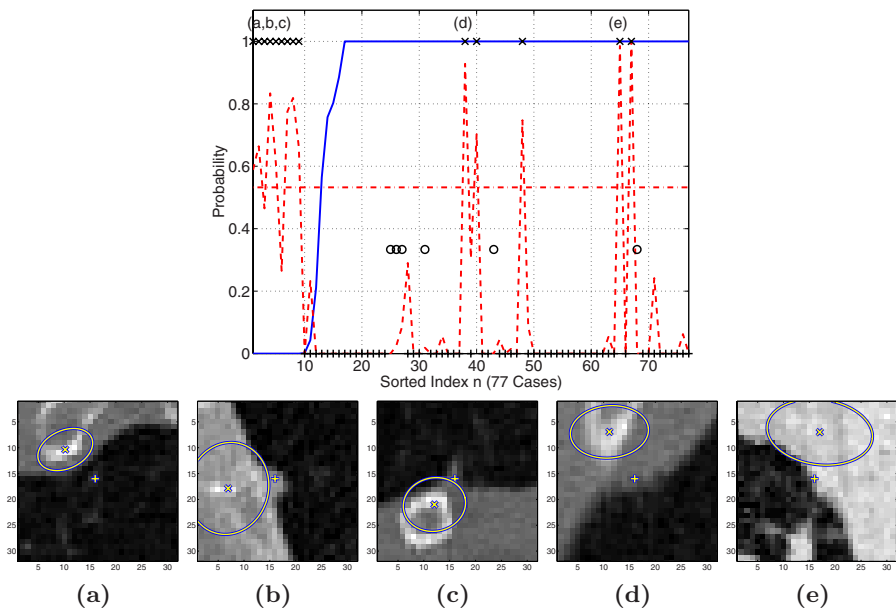


Fig. 5. Experimental results for the validation process. The top plot illustrates the Q probability (solid-line) and β estimate (dashed-line) for each test case. The symbols “+”, “x”, and “o” indicate correct, failure, and GGO nodule cases, respectively. β values are normalized to fit within the range of this plot. A horizontal dashed-line indicates the β -threshold $th_2 = 400$. The bottom images show examples of correctly rejected failures. Legend of these images are the same as Fig.4. Cases (a) and (c) satisfied the rejection conditions of both Q and β while Case (b) met only the Q condition and Cases (d) and (e) met only the β condition.

failures were due to small tumors whose shape was a partial ellipsoid located on lung walls and near rib structures. All the 14 failures were successfully rejected by the validation process without false rejection and false acceptance. The data includes six cases of the part- and non-solid or ground-glass opacity nodules (GGO nodules, see Fig.1c,d and Fig.4a,b). All GGO nodules were successfully estimated and accepted.

Fig.4 shows examples of the resulting center and spread estimates. It illustrates cases with the irregular, GGO, and on-the-wall nodules whose geometrical shapes are largely deviated from the Gaussian structure. The correct estimations for these difficult cases demonstrate the robustness and effectiveness of our framework. Fig.5 shows the results of the statistical validation and examples of the rejected cases. In order to evaluate the generalization capability, we apply the same validation process to different lung HRCT data of 3 patients captured in different settings. This preliminary study resulted in 96% correct validation rate (4 false acceptances among 100 trials), similar to the results shown in Fig.5.

5 Conclusions

This paper proposed a robust estimation and validation framework for characterizing the location and anisotropic spread of local data structure that is approximated by a Gaussian-based model. The new framework unifies the mean shift-based robust statistical estimation and the linear scale-space-based multi-scale analysis. The unification is realized by formally extending the mean shift-based analysis towards the evaluation of continuous positive function whose local structure is characterized by an anisotropic fully-parameterized covariance matrix. The proposed statistical validation method also complements this estimation framework, providing an effective goodness-of-fit measure for rejecting accidental ill-estimates. The strength of our solution is its robustness against the margin-truncation and the non-Gaussianity effects. This advantage was demonstrated by the experimental results with the 1D and 2D synthetic data and by the 3D tumor analysis application.

Our proposed method can be interpreted as a multi-scale joint Gaussian fitting and segmentation. The estimation scheme achieves fitting by using only samples within the basin of attraction for characterizing the underlying structure. The importance of considering the anisotropic covariance was also suggested by Lillholm et al. [11] in their image reconstruction analyses with various local features defined as combinations of the first and second order derivatives of the scale-space representations. Their results have direct implications to our problem since the second order derivatives (or Hessian matrix) are explicitly related to the covariance matrix [13, p.178][8].

The results with the real lung HRCT 3D data demonstrated a successful application of our method to the volumetric tumor analysis, providing accurate estimation of 3D location and anisotropic spread of the non-spherical pulmonary tumors. The robustness and flexibility facilitates not only the medical applications sought in this paper but also various other applications involving with the analysis of blob-like data structures. A natural continuation of this study is the extension of our framework for the automatic tumor detection problem. This remains as our future work.

Acknowledgments. The authors wish to thank Visvanathan Ramesh from Siemens Corporate Research for stimulating discussions, Alok Gupta from CAD group, Siemens Medical Solutions, for his support and encouragement, and Jonathan Stoeckel from CAD group, Siemens Medical Solutions, for his valuable technical supports.

References

1. Y. Chen and J. McInroy. Estimating symmetric, positive definite matrices in robotic control. In *IEEE Int. Conf. Robotics and Automation*, pages 4269–4274, Washington D.C., 2002.
2. D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(2):281–288, 2003.
3. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(5):603–619, 2002.
4. K. Fukunaga. *Statistical Pattern Recognition*. Academic Press, San Diego, 1990.
5. G. Golub and C. van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1996.
6. C. Henschke, D. Yankelevitz, R. Mirtcheva, G. McGuinness, and O. McCauley, D. Miettinen. CT screening for lung cancer: frequency and significance of part-solid and nonsolid nodules. *AJR Am. J. Roentgenol.*, 178(5):1053–1057, 2002.
7. H. Hu. Positive definite constrained least-squares estimation of matrices. *Linear Algebra and Its Applications*, 229:167–174, 1995.
8. Y. Kanazawa and K. Kanatani. Do we really have to consider covariance matrices for image features? In *Int. Conf. Computer Vision*, pages 586–591, Vancouver, 2001.
9. J. Koenderink. The structure of images. *Biol. Cybern.*, 50:363–370, 1984.
10. Y. Lee, T. Hara, H. Fujita, S. Itoh, and T. Ishigaki. Automated detection of pulmonary nodules in helical CT images based on an improved template-matching technique. *IEEE Trans. Medical Imaging*, 20(7):595–604, 2001.
11. M. Lillholm, M. Nielsen, and L. Griffin. Feature-based image analysis. *Int. J. Comput. Vision*, 52(2/3):73–95, 2003.
12. T. Lindeberg. Feature detection with automatic scale selection. *Int. J. Comput. Vision*, 30(2):79–116, 1998.
13. B. Matei. *Heteroscedastic Errors-In-Variables Models in Computer Vision*. PhD thesis, Rutgers University, 2001.
14. M. Nielsen, L. Florack, and R. Deriche. Regularization, scale space, and edge detection filters. *J. Mathematical Imaging and Vision*, 7(4):291–307, 1997.
15. P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(7):629–639, 1990.
16. W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 1992.
17. P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. John Wiley, New York, 1987.
18. H. Takizawa, S. Yamamoto, T. Matsumoto, Y. Tateno, T. Iinuma, and M. Matsumoto. Recognition of lung nodules from X-ray CT images using 3D markov random field models. In *Int. Conf. Pattern Recog.*, Quebec City, 2002.
19. B. van Ginneken, B. ter Harr Romeny, and M. Viergever. Computer-aided diagnosis in chest radiography: A survey. *IEEE Trans. Medical Imaging*, 20(12):1228–1241, 2001.
20. S. van Huffel and J. Vandewalle. *The Total Least Squares Problem Computational Aspects and Analysis*. SIAM, Philadelphia, 1991.
21. A. Witkin. Scale-space filtering. In *Int. Joint. Conf. Artificial Intell.*, pages 1019–1021, Karlsruhe, 1983.

Dimensionality Reduction by Canonical Contextual Correlation Projections

Marco Loog¹, Bram van Ginneken¹, and Robert P.W. Duin²

¹ Image Sciences Institute, University Medical Center Utrecht
Utrecht, The Netherlands
`{marco, bram}@isi.uu.nl`

² Information and Communication Theory Group, Delft University of Technology
Delft, The Netherlands
`r.p.w.duin@ewi.tudelft.nl`

Abstract. A linear, discriminative, supervised technique for reducing feature vectors extracted from image data to a lower-dimensional representation is proposed. It is derived from classical Fisher linear discriminant analysis (LDA) and useful, for example, in supervised segmentation tasks in which high-dimensional feature vector describes the local structure of the image. In general, the main idea of the technique is applicable in discriminative and statistical modelling that involves contextual data.

LDA is a basic, well-known and useful technique in many applications. Our contribution is that we extend the use of LDA to cases where there is dependency between the output variables, i.e., the class labels, and not only between the input variables. The latter can be dealt with in standard LDA.

The principal idea is that where standard LDA merely takes into account a single class label for every feature vector, the new technique incorporates class labels of its neighborhood in its analysis as well. In this way, the spatial class label configuration in the vicinity of every feature vector is accounted for, resulting in a technique suitable for e.g. image data. This spatial LDA is derived from a formulation of standard LDA in terms of canonical correlation analysis. The linearly dimension reduction transformation thus obtained is called the canonical contextual correlation projection.

An additional drawback of LDA is that it cannot extract more features than the number of classes minus one. In the two-class case this means that only a reduction to one dimension is possible. Our contextual LDA approach can avoid such extreme deterioration of the classification space and retain more than one dimension.

The technique is exemplified on a pixel-based segmentation problem. An illustrative experiment on a medical image segmentation task shows the performance improvements possible employing the canonical contextual correlation projection.

1 Introduction

This paper describes a supervised technique for linearly reducing the dimensionality of image feature vectors (e.g. observations in images describing the local gray level structure at certain positions) taking contextual label information into account (e.g.

the local class label configuration in a segmentation task). The technique is based on canonical correlation analysis and called the canonical contextual correlation projection (CCCP).

In general, the main goal of reducing the dimensionality of feature data is to prevent the subsequently used model from over-fitting in the training phase [9,12]. An important additional effect in, for example, pattern classifiers is often the decreased amount of time and memory required to perform the necessary operations. Consequently image segmentation, object classification, object detection, etc. may benefit from the technique, and also other discriminative methods may gain from it.

The problem this paper is concerned with is of great practical importance within real-world, discriminative and statistical modelling tasks, because in many of these image analysis and computer vision tasks the dimensionality, say n , of the feature data can be relatively large. For example, because it is not clear a priori what image information is needed for a good performance in a pixels classification task, many features per pixel may be included, which results in a high-dimensional feature vector. This already happens in 2-dimensional image processing, but when processing large hyper-spectral images, medical 3-dimensional volumes, or 4-dimensional space/time image data, it may even be less clear what features to take and consequently more features are added. However, high-dimensional data often leads to inferior results due to the curse of dimensionality [4, 12] even if all relevant information for accurate classification is contained in the feature vector. Hence, lowering the dimensionality of the feature vectors can lead to a significant gain in performance.

The CCCP is an extension to linear discriminant analysis (LDA), which is a well-known supervised dimensionality reduction technique from statistical pattern recognition [9,12]. LDA is capable of taking contextual information in the input variables into account, however contextual information in the output variables is not explicitly dealt with. The CCCP does take this information into account and therefore models this contextual information more accurately.

Another principal drawback of LDA is that it cannot extract more features than the number of classes minus one [7,9]. In the two-class case—often encountered in image segmentation or object detection—this means that we can only reduce the dimensionality of the data to one, and even though reducing the dimensionality could improve the performance it is not plausible that one single feature can describe class differences accurately. CCCP can avoid such extreme deterioration of the classification space and retain more than one dimension even in the case of two-class data.

LDA was originally proposed by Fisher [5,6] for the two-class case and extended by Rao [14] to the multi-class case. The technique is supervised, i.e., input and output patterns which are used for training have to be provided. Quite a few linear dimension reduction techniques have been proposed of which many are variations and extensions to Fisher's LDA, see [3,9,16]. Within the field of image classification, [1] and [13] show how classification performance can benefit from linear dimension reduction. The novel extension to LDA given in this paper explicitly deals with the contextual spatial characteristics of image data. To come to this extension of LDA, a formulation of this technique in terms of canonical correlation analysis (CCA, [11]) is used (see [9,16]), which enables us to not only to include the class labels of the pixel that is considered—as in classical LDA, but also to encode information from the surrounding class labelling

structure. Related to our approach is the work of Borga [2] in which CCA is also used as a framework for image analysis.

Finally, it is mentioned that there is a close relationship of the LDA considered here and a form of LDA which is used for classification. The latter is also known as a linear discriminant classifier or Fisher's linear discriminant [9,16]. Here, however, LDA for dimensionality reduction is considered.

1.1 Outline

Section 2 formulates the general problem statement within the context of supervised image segmentation. However, we stress that the technique is not restricted to this task. Techniques like object detection or object classification can also benefit from the dimension reduction scheme proposed. Section 3 introduces LDA and discusses its link to CCA. Subsection 3.4 presents the CCCP. Subsection 3.5 discusses the drawback of obtaining too few dimensions with LDA, and explains how CCCP can overcome this limitation. Subsection 3.6 summarizes the main approach. Section 4 presents illustrative results on a lung field segmentation task in chest radiographs. Finally, Section 5 provides a discussion and conclusions.

2 Problem Statement

To make the exposition more clear, the technique presented is directly related to the specific task of image segmentation, and it is not discussed in its full generality.

An image segmentation task in terms of pixel classification is considered—however, we may as well use other image primitives on a regular lattice. Based on image features associated to a pixel, it is decided to which of the possible classes this pixel belongs. Having classified all pixels in the image gives a segmentation of this image. Examples of features associated to a pixel are its gray level, gray levels of neighboring pixels, texture features, the position in the image, gray levels after linear or non-linear filtering of the image, etc.

Pixels are denoted by p_i and the features extracted from the image associated to p_i are represented in an n -dimensional feature vector \mathbf{x}_i . A classifier maps \mathbf{x}_i to a certain class label coming from a set of K possibilities $\{l_1, \dots, l_K\}$. All pixels having the same label belong to the same segment and define the segmentation of the image. The classifier, e.g. a quadratic classifier, Fisher's linear discriminant, a support vector machine, or a k nearest neighbor classifier [9,12], is constructed using train data: example images and their associated segmentations should be provided beforehand from which the classifier learns how to map a given feature vector to a certain class label.

Before training the classifier, a reduction of dimensionality can be performed using the train data. This is done by means of a linear projection \mathbf{L} from n to d ($d < n$) dimensions, which can be seen as a $d \times n$ -matrix that is applied to the n -dimensional feature vectors \mathbf{x}_i to get a d -dimensional feature representation $\mathbf{L}\mathbf{x}_i$. The matrix \mathbf{L} is determined using the train data. Subsequently, the feature vectors of the train data are transformed to the lower dimensional feature vectors and the classifier is constructed using these transformed feature vectors. The following section presents a novel way to determine such a matrix \mathbf{L} .

3 Canonical Contextual Correlation Projections

3.1 Linear Discriminant Analysis

The classical approach to supervised linear dimensionality reduction is based on LDA. This approach defines the optimal transformation matrix \mathbf{L} to be the one that maximizes the so-called Fisher criterion J

$$J(\mathbf{L}) = \text{tr}((\mathbf{L}\mathbf{S}_W\mathbf{L}^t)^{-1}\mathbf{L}\mathbf{S}_B\mathbf{L}^t), \quad (1)$$

where \mathbf{L} is the $d \times n$ transformation matrix, \mathbf{S}_W is the mean within-class covariance matrix, and \mathbf{S}_B is the between-class covariance matrix. The $n \times n$ -matrix \mathbf{S}_W is a weighted mean of class covariance matrices and describes the (co)variance that is (on average) present within every class. The $n \times n$ -matrix \mathbf{S}_B describes the covariance present between the several classes. In Equation (1), $\mathbf{L}\mathbf{S}_W\mathbf{L}^t$ and $\mathbf{L}\mathbf{S}_B\mathbf{L}^t$ are the $d \times d$ within-class and between-class covariance matrices of the feature data after reducing the dimensionality of the data to d using the linear transform \mathbf{L} .

When maximizing (1), one simultaneously minimizes the within-class covariance and maximizes the between-class covariance. The criterion tries to determine a transform \mathbf{L} that maps the feature vectors belonging to one and the same class as close as possible to each other, while trying to keep the vectors that do not belong to the same class as far from each other as possible. The matrix that does so in the optimal way, as defined by (1), is the transform associated to LDA.

Once the covariance matrices \mathbf{S}_W and \mathbf{S}_B have been estimated from the train data, the maximization problem in (1) can be solved by means of a generalized eigenvalue decomposition involving the matrices \mathbf{S}_B and \mathbf{S}_W . We do not discuss these procedures here, but refer to [3,4,7] and [9].

3.2 Canonical Correlation Analysis

This paper formulates LDA in a canonical correlation framework (see [9,16]) which enables the extension of LDA to CCCP. CCA is a technique to extract, from two feature spaces, those lower-dimensional subspaces that exhibit a maximum mutual correlation [11,2].

To be more precise, let X be a multivariate random variable, e.g. a feature vector, and let Y be another multivariate random variable, e.g. a numeric representation of the class label: $(1, 0, \dots, 0)^t$ for class 1, $(0, 1, \dots, 0)^t$ for class 2, etc. In addition, let \mathbf{a} and \mathbf{b} be vectors (linear transformations) having the same dimensionality as X and Y , respectively. Furthermore, define c to be the correlation between the univariate random variables $\mathbf{a}^t X$ and $\mathbf{b}^t Y$, i.e.,

$$c = \frac{E(\mathbf{a}^t X \mathbf{b}^t Y)}{\sqrt{E((\mathbf{a}^t X)^2)E((\mathbf{b}^t Y)^2)}}, \quad (2)$$

where E is the expectation. The first canonical variates $\mathbf{a}_1^t X$ and $\mathbf{b}_1^t Y$ are obtained by those two vectors \mathbf{a}_1 and \mathbf{b}_1 that maximize the correlation in Equation (2). The second canonical variates are those variates that maximize c under the additional constraint that

they are outside the subspace spanned by \mathbf{a}_1 and \mathbf{b}_1 , respectively. Having the first two pairs of canonical variates, one can construct the third, by taking them outside the space spanned by $\{\mathbf{a}_1, \mathbf{a}_2\}$ and $\{\mathbf{b}_1, \mathbf{b}_2\}$, etc.

One way of solving for the canonical variates more easily is as follows. First estimate the matrices \mathbf{S}_{XX} , \mathbf{S}_{YY} , and \mathbf{S}_{XY} , that describe the covariance for the random variables X and Y , and the covariance between these variables, i.e., estimating $E(XX^t)$, $E(YY^t)$, and $E(XY^t)$, respectively. Subsequently, determine the eigenvectors \mathbf{a}_i of

$$\mathbf{S}_X := \mathbf{S}_{XX}^{-1} \mathbf{S}_{XY} \mathbf{S}_{YY}^{-1} \mathbf{S}_{XY}^t \quad (3)$$

and the \mathbf{b}_j of

$$\mathbf{S}_Y = \mathbf{S}_{YY}^{-1} \mathbf{S}_{XY}^t \mathbf{S}_{XX}^{-1} \mathbf{S}_{XY}. \quad (4)$$

The two eigenvectors \mathbf{a}_1 and \mathbf{b}_1 associated with the largest eigenvalues of the matrices \mathbf{S}_X and \mathbf{S}_Y , respectively, are the vectors giving the first canonical variates $\mathbf{a}_1^t X$ and $\mathbf{b}_1^t Y$. For the second canonical variates take the eigenvectors \mathbf{a}_2 and \mathbf{b}_2 with the second largest eigenvalues associated, etc. The number of canonical variates that can be obtained is limited by the smallest rank of both multivariate random variables considered.

3.3 LDA through CCA

LDA can be defined in terms of CCA (see for example [9] or [16]), hence avoiding the use of the Fisher criterion (1). To do so, let X be the random variable describing the feature vectors and let Y describe the class labels. Without loss of generality, it is assumed that X is centered, i.e., $E(X)$ equals the null vector. Furthermore, as already suggested in Subsection 3.2, the class labels are numerically represented as K -dimensional standard basis vectors: for every class one basis vector.

Performing CCA on these random variables using \mathbf{S}_X from (3), one obtains eigenvectors \mathbf{a}_i that span the space (or part of this space) of n -dimensional feature vectors. A transformation matrix \mathbf{L} , equivalent to the one maximizing the Fisher criterion, is obtained by taking the d eigenvectors associated to the d largest eigenvalues and putting them as row-vectors in the transformation matrix:

$$\mathbf{L} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_d)^t.$$

Linear dimensionality reduction performed with this transformation matrix gives results equivalent to classical LDA. Note that to come to this solution, an eigenvalue decomposition of \mathbf{S}_Y is not needed.

The estimates of the covariance matrices used later on in our experiments are the well-known maximum likelihood estimates. Given N pixels p_i in our train data set, and denoting the numeric class label representation of pixel p_i by the K -dimensional vector \mathbf{y}_i , \mathbf{S}_{XX} is estimated by the matrix

$$\frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \mathbf{y}_i^t.$$

\mathbf{S}_{XX} and \mathbf{S}_{YY} are estimated in a similar way.

The CCA formulation of LDA enables us to extend LDA to a form of correlation analysis that takes the spatial structure of the class labelling in the neighborhood of the pixels into account.

3.4 Incorporating Spatial Class Label Context

In image processing, incorporating spatial gray level context into the feature vector is readily done by not only considering the actual gray level in that pixel as a feature, but by taking additional gray levels of neighboring pixels into account, or by adding large-scale filter outputs to the feature vector. However, on the class label side there is also contextual information available. Although two pixels could belong to the same class—and thus have the same class label, the configuration of class labels in their neighborhood can differ very much. LDA and other dimension reduction techniques, do not take into account this difference in spatial configuration, and only consider the actual label of the pixel.

The trivial way to incorporate these differences into LDA would be to directly distinguish more than K classes on the basis of these differences. Consider for example the 4-neighborhood label configurations in Figure 1. In a $K = 2$ -class case, this 4-neighborhood could attain a maximum of $2^5 = 32$ different configurations (of which only four are displayed in the figure). These could then be identified as being different classes. Say we have M of them, then every configuration possible would get its own unique M -dimensional standard basis vector (as in Subsection 3.3) and one could subsequently perform LDA based on these classes, in this way indirectly taking more than a single class label into account when determining a dimension reducing matrix \mathbf{L} .

However, identifying every other configuration with a different class seems too crude. When two neighborhood label configurations differ in only a single pixel label, they should be considered more similar to each other than two label configurations differing in half of their neighborhood. Therefore, in our CCCP approach, a class label vector \mathbf{y}_i is not encoded as a null vector with a single one in it, i.e., a standard basis vector, but as a 0/1-vector in which the central pixel label and every neighboring label is encoded as a K -dimensional (sub)vector. Returning to our 2-class example from Figure 1, the

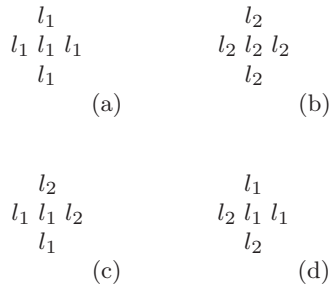


Fig. 1. Four possible class labellings in case a four-neighborhood context is considered. For this two-class problem the total number of possible contextual labellings equals $2^5 = 32$.

four label vectors that would give the proper encoding of the class labelling within these 4-neighborhoods (a), (b), (c), and (d) are

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \text{ and } \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}. \quad (5)$$

The five pixels (the four pixels in the neighborhood and the central pixel) are traversed left to right and top to bottom. So the first two entries of the four vectors correspond to the labelling of the top pixel and the last two entries correspond to the bottom pixel label.

Note that the label vectors are 10-dimensional, i.e., per pixel from the neighborhood (five in total) a sub-vector of size two is used to encode the two possible labellings per pixel. In general, if P is the number of pixels in the neighborhood including the central pixel, these (KP) -dimensional vectors contain P ones, and $(K - 1)P$ zeros, because every pixel belongs to exactly one of K classes, and every pixels is thus represented by a K -dimensional sub-vector. In the foregoing example where $K = 2$ and $P = 5$, there are 5 ones and 5 zeros in the complete vector, and 1 one and 1 zero per sub-vector.

When taking the contextual label information into account in this way, gradual changes in the neighborhood structure are appreciated. In Figure 1, configurations (a) and (b) are as far from each other as possible (in terms of e.g. Euclidean or Hamming distance, cf. the vectors in (5)), because in going from one configuration to the other, all pixel sites have to change their labelling. Comparing a different pair of labellings from Figure 1 to each other, we see that their distance is less than maximal, because it needs less permutations to turn one contextual labelling into the other.

We propose the numeric class label encoding described above for incorporating contextual class label information into the CCA, resulting in the canonical correlation projection, CCCP, that can explicitly deal with gray value context—through the feature vectors \mathbf{x}_i —as well as with class label context—through our numeric class label encoding represented by the vectors \mathbf{y}_i . Note that CCCP encompasses classical LDA. Taking no class label context into account but only the class label of the central pixel clearly reduces CCCP to LDA.

3.5 Reduction to More than $K - 1$ Dimensions

We return to one of the main drawbacks of LDA already mentioned: the fact that LDA cannot reduce the dimensionality to more than $K - 1$, i.e., the number of classes minus 1. In many segmentation tasks K is not higher than 2 or 3, in which case LDA can only

extract 1 or 2 dimensions. Starting with a high-dimensional image feature space, it is hardly to be expected that all relevant information is captured in this subspace.

The CCCP alleviates this limitation. The maximum number of canonical variates that can be extracted through CCA equals $\min\{\text{rank}(\mathbf{S}_X), \text{rank}(\mathbf{S}_Y)\}$. When dealing with as many as or fewer classes than the feature dimensionality, i.e., $K \leq n$, the limiting factor in the dimensionality reduction using LDA is the matrix \mathbf{S}_Y which rank is equal to (or even smaller than) $K - 1$. However, by extending the class label context, the rank of \mathbf{S}_Y increases and can even get larger than $\text{rank}(\mathbf{S}_X)$.

So in general, CCCP can provide more canonical variates than classical LDA by incorporating more class label context. And consequently, for CCCP the resulting feature dimensionality can be larger than $K - 1$. In the experiments in Section 4, it is shown that this can significantly improve the segmentation results.

3.6 The CCCP Algorithm

The CCCP technique is summarized. A reduction of n -dimensional image data to d dimensions is considered.

- define what (contextual) image feature information to use (e.g. which filters), and which neighboring pixels to take for the class label context
- determine from the train images and associated segmentations the gray level feature vectors \mathbf{x}_i
- determine from the same data the class label feature vectors \mathbf{y}_i , i.e., determine for every pixel in the context its standard basis vector describing its class label and concatenate all these vectors
- estimate the covariance matrices \mathbf{S}_{XX} , \mathbf{S}_{XY} , and \mathbf{S}_{YY}
- do an eigenvalue decomposition of the matrix $\mathbf{S}_X := \mathbf{S}_{XX}^{-1} \mathbf{S}_{XY} \mathbf{S}_{YY}^{-1} \mathbf{S}_{XY}^t$ from (3)
- take the d rows of the $d \times n$ linear dimension reducing transformation matrix \mathbf{L} equal to the d eigenvectors associated to the d largest eigenvalues
- transform all \mathbf{x}_i using \mathbf{L} to \mathbf{Lx}_i

4 Illustrative Experiments

This section exemplifies the theory by a simple illustrative example. The section is not intended to present a full-fledged state-of-the-art solution to the task, but merely to illustrate the possible improvements in performance when employing the CCCP instead of the original LDA or no dimensionality reduction at all. For this reason, the task considered is a lung field segmentation task in chest radiographs, which is based on a simple pixel classification technique. A segmentation scheme solving this problem properly may be based on snakes, active shape models, or some kind of Markov random field, taking more global contextual and/or geometric information into account (cf. [8]).

4.1 Chest Radiograph Data

The data used in the experiments consists of 20 standard PA chest radiographs taken from a tuberculosis screening programm. The size of the sub-sampled and digitized images

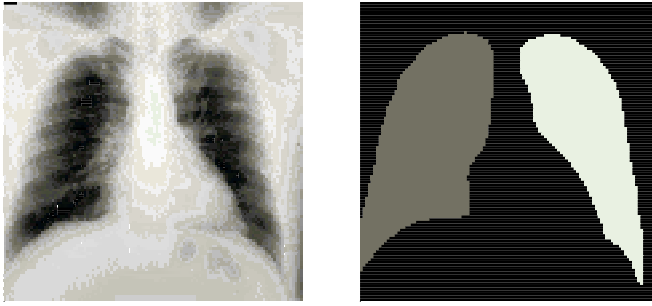


Fig. 2. The left image displays a typical PA chest radiograph as used in our experiments. The right image shows its expert lung field segmentation. The background is black, both lung fields are in different shades of gray.

equals 128×128 . An examples of a typical chest radiographs is shown in Figure 2. The task is to segment, both lung fields.

In addition to the radiographs, the associated ground truth is given, i.e., in these images, the the lung fields are manually delineated by an expert and the delineation is converted to a 3-class pixel labelling. An example image is given in Figure 2 also.

4.2 Experimental Setup

In all experiment, 10 images were used for training and 10 for testing. The total number of feature vectors equals $20 \cdot (128 - 12)^2 = 269,120$ and train and test set both contain half of it. Note that pixel within a distance of 6 pixels from the border are not taken into account to avoid boundary problems in building up the contextual gray level features (see below).

Experiments were conducted using a nonparametric 1 nearest neighbor (1NN) classifier. We chose to use a 1NN classifier for its simplicity and because it offers suitable baseline results which makes a reasonable comparison possible [3,7,12]. Before the 1NN classifier was trained, the within-class covariance matrix \mathbf{S}_W was whitened (cf. Subsection 3.1) based on the train data [7].

The variables in our experiments were the contextual class label information, and the dimensionality d to which the data is to be reduced. The contextual class label information belonging to one pixel p_i is defined by all pixels coming from within a radius of r pixels from p_i . Experiments were performed with $r \in \{0, 2, 3\}$; $r = 0$ means that only the central label belonging to p_i is taken into account (equal to classical LDA), $r = 2$ results in 13 contextual labels, and $r = 3$ in 29 contextual labels.

As contextual image features, we simply took the gray levels from neighboring pixels into account, so no filtering or other preprocessing is performed. The contextual information of pixel p_i consisted of all raw gray values within a radius of 5 from this pixel. In addition, the x and y coordinates were added to the image feature vector, which final dimensionality totals $81 + 2 = 83$. (Choosing to set the radius for the contextual gray level information to 5 is based on a small pilot experiment using LDA. LDA performed best with these settings.)

The dimensionality d to reduce to were in the set $\{1, 2, 4, 11, 22, 37, 56, 83\}$. N.B. setting d equal to 83 means no dimensionality reduction is performed.

Using the aforementioned d , image features and contextual class label features, the train set was used for determining the CCCP and training the 1NN classifier. Subsequently, using the test set, we determined the pixel classification error.

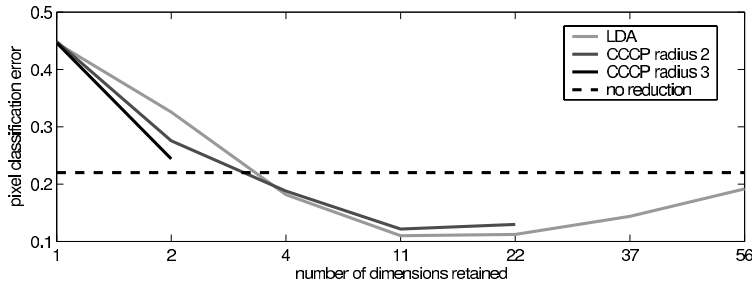


Fig. 3. The black dashed horizontal line indicates the performance of the pixel classification scheme if no dimensionality reduction is employed and the full 83-dimensional feature vector is used in the segmentation. The black solid line is the classification error obtained when using LDA. The gray lines give the performance for the two different instances of CCCP. The dark gray line uses a contextual radius of 2, while the light gray line uses a radius of 3. Their pixel classification error is plotted against feature dimensionality d . The optimal classification errors are 0.22, 0.24, 0.12, and 0.11, respectively.

4.3 Results

Figure 3 gives the results obtained by LDA, CCCP and no dimensionality reduction. Note that for LDA (solid black line), the dimensionality can only be reduced to 1 or 2, because the number of classes K is 3 (i.e., left lung field, right lung field, or background). Note also the peaking behavior [4,12] that is visible in the plots of the CCCP results.

Both instances of CCCP clearly outperforms LDA and they give a dramatic improvement over performing no dimensionality reduction as well. It should be noted, though, that CCCP does not outperform LDA for every dimensionality d .

Figure 4 gives for the example image in Figure 2 the segmentation obtained by the optimal LDA (left), the segmentation obtained by the optimal CCCP (middle), and the one obtained using no reduction (right). Comparing the three images, the main observations is that the CCCP-based segmentation gives much more coherent results than the other segmentations. Furthermore, there seems to be less confusion between left and right lung fields when CCCP is employed. The background classification error in comparison with the result in the right image, however, seems to go up a bit when using the CCCP approach. In the right image there are no misclassified background pixels, in both other images there are.

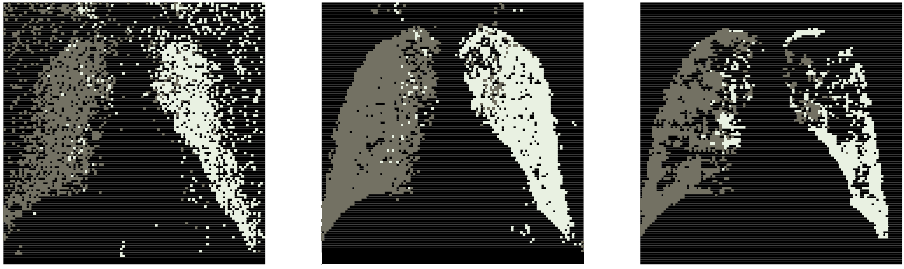


Fig. 4. The segmentation with optimal LDA ($d = 2$) is depicted on the left, the one with optimal CCCP in the middle ($d = 11$ and $r = 3$), and on the right is the segmentation obtained using no dimensionality reduction.

5 Discussion and Conclusions

In this work we extended classical LDA—as a dimensionality reduction technique—to incorporate the spatial contextual structure present in the class labelling. Our extension, called the canonical contextual correlation projection (CCCP), is based on a canonical correlation formulation of LDA that enables the encoding of these spatial class label configurations. Experiments on the task of segmenting the lung fields in chest radiographs demonstrated that in this way significant improvement over LDA or no dimension reduction is possible. Furthermore, these experiments show also that using a data-driven method for image segmentation—in which the dimension reduction is an essential part, good results can be obtained without the additional utilization of task-dependent knowledge. We expect that similar results hold in, for example, object detection, object classification or some other discriminative tasks in which CCCP can also be used to determine low-dimensional but still discriminative features.

Clearly, regarding the experiments, improving the segmentation results should be possible. For example, by using more complex pattern recognition techniques that can also handle contextual class label information in their classification scheme. Typically, such scheme employs a Markov random field approach, or something closely resembling this [10,15,17]. Here CCCP could also be a valuable tool in another way. Due to the iterative nature of these schemes they often are rather slow. In part, this may be attributed to the large contextual neighborhoods that are taken into account. Lowering the dimensionality of these neighborhoods can, in addition to improving the error rate, speed up the iterative process considerably.

An interesting way to further improve the dimensionality reduction scheme is the development of nonlinear CCCP. This is for example possible via a CCA-related technique called optimal scoring [9], which is, among other things, used for extending LDA to nonlinear forms. Nonlinear dimensionality reduction can of course lead to a better lower-dimensional representation of the image data, however the nonlinearity often makes such approaches computationally hard. Nonetheless, CCCP does (via CCA) provide a proper framework for these kind of extensions.

In conclusion, CCCP provides a general framework for linearly reducing contextual feature data in a supervised way, it is well capable of improving LDA and can be ex-

tended in several directions. It generalizes LDA by not only taking gray level context into account, but incorporating contextual class label information as well. In a small segmentation experiment, it was shown that CCCP can clearly give improvement performance compared to LDA and no dimensionality reduction.

References

1. P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.
2. M. Borga. *Learning Multidimensional Signal Processing*. Ph.D. Thesis, Linköping University, Sweden, 1998.
3. P. A. Devijver and J. Kittler. *Pattern Recognition: a Statistical Approach*. Prentice-Hall, London, 1982.
4. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, New York, second edition, 2001.
5. R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.
6. R. A. Fisher. The statistical utilization of multiple measurements. *Annals of Eugenics*, 8:376–386, 1938.
7. K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 1990.
8. B. van Ginneken, B. M. ter Haar Romeny, and M. A. Viergever. Computer-aided diagnosis in chest radiography: A survey. *IEEE Transactions on Medical Imaging*, 20(12):1228–1241, 2001.
9. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer-Verlag, New York . Berlin . Heidelberg, 2001.
10. N. L. Hjort and E. Mohn. A comparison in some contextual methods in remote sensing classification. In *Proceedings of the 18th International Symposium on Remote Sensing of Environment*, pages 1693–1702, Paris, France, 1984. CNES.
11. H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.
12. A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.
13. K. Liu, Y.-Q. Cheng, and J.-Y. Yang. Algebraic feature extraction for image recognition based on an optimal discriminant criterion. *Pattern Recognition*, 26(6):903–911, 1993.
14. C. R. Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B*, 10:159–203, 1948.
15. J. A. Richards, D. A. Landgrebe, and P. H. Swain. Pixel labeling by supervised probabilistic relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(2):188–191, 1981.
16. B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, 1996.
17. G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*. Number 27 in Applications of mathematics. Springer-Verlag, Berlin . Heidelberg, 1995.

Accuracy of Spherical Harmonic Approximations for Images of Lambertian Objects under Far and Near Lighting

Darya Frolova, Denis Simakov, and Ronen Basri

Dept. of Computer Science and Applied Math,
The Weizmann Institute of Science,
Rehovot 76100, Israel,
{darya.frolova,denis.simakov,ronen.basri}@weizmann.ac.il

Abstract. Various problems in Computer Vision become difficult due to a strong influence of lighting on the images of an object. Recent work showed analytically that the set of all images of a convex, Lambertian object can be accurately approximated by the low-dimensional linear subspace constructed using spherical harmonic functions. In this paper we present two major contributions: first, we extend previous analysis of spherical harmonic approximation to the case of *arbitrary objects*; second, we analyze its applicability for *near light*. We begin by showing that under distant lighting, with uniform distribution of light sources, the average accuracy of spherical harmonic representation can be bound from below. This bound holds for objects of arbitrary geometry and color, and for general illuminations (consisting of any number of light sources). We further examine the case when light is coming from above and provide an analytic expression for the accuracy obtained in this case. Finally, we show that low-dimensional representations using spherical harmonics provide an accurate approximation also for fairly near light. Our analysis assumes Lambertian reflectance and accounts for attached, but not for cast shadows. We support this analysis by simulations and real experiments, including an example of a 3D shape reconstruction by photometric stereo under very close, unknown lighting.

1 Introduction

Methods for solving various Computer Vision tasks such as object recognition and 3D shape reconstruction under realistic lighting often require a tractable model capable of predicting images under different illumination conditions. It has been shown in [3] that even for the simple case of Lambertian (matt) objects the set of all images of an object under varying lighting conditions occupies a volume of unbounded dimension. Nevertheless many researchers observed that in many practical cases this set lies close to a low-dimensional linear subspace [4,6,18]. Low-dimensional representations have been used for solving many Computer Vision problems (e.g., [8,10,17]).

Low dimensional representations of lighting have been recently justified analytically in [1,14]. These studies show that the set of all Lambertian reflectance

functions (the mapping from surface normals to intensities) is, to an accurate approximation, low dimensional, and that this space is spanned by the low order *spherical harmonics*. Explicit spherical harmonic bases have been used to solve a number of important problems: object recognition [1], photometric stereo [2], reconstruction of moving shapes [16], and image rendering [15].

The introduction of spherical harmonic analysis provides a useful tool for handling complex illumination, but this pioneering work [1,14] is incomplete in a practically important aspect: the analysis in [1,14] is not easily generalized for the case of arbitrary object shapes and albedos.

In this paper we consider the case of Lambertian reflectance allowing for attached, but not for cast shadows. Thus, our analysis is applicable to convex objects illuminated by arbitrary combinations of point and extended sources. We begin by showing that under distant lighting the average accuracy of spherical harmonic representations can be bound from below by a bound that is *independent of the shape* of the object. For this result we assume that lighting can be cast on an object from any direction with equal probability, and that the distribution of the intensity of lighting is independent of its direction. We further consider a second case in which lighting is illuminating the object only from above, and derive an expression that allows us to calculate the accuracy of the spherical harmonic representation in this case.

While we consider a *single* expression for the harmonic basis there are studies that seek to build an *optimal basis* for every specific object or illumination. Ramamoorthi in [13] presents analytical construction of an optimal basis for the space of images. His analysis is based on spherical harmonics, and the images are taken under point light sources (uniformly distributed). The results of [13] are generalized and extended in [11,12] for different illumination distributions and materials. While they consider specific object geometries, our goal is to bound from below the approximation accuracy for arbitrary objects.

In the second part of our paper we analyze what happens if we relax the assumption of infinitely distant illumination, and show that spherical harmonics still provide a good basis even for fairly close light. We find what distance to the light can be considered infinite, as far as a spherical harmonic approximation is concerned. Our results show that although the approximation accuracy can be very bad for extremely close light, it rapidly increases as the distance to the light grows and even at rather small distances we achieve quite a good accuracy.

The assumption of infinitely distant light greatly simplifies the analysis of illumination effects, and so it is widely utilized in Computer Vision studies. While there are studies that incorporate near light effects (as in [7]), we are unaware of previous theoretical analysis of this factor.

The paper is divided as follows. In Section 2 we briefly review the use of spherical harmonics to represent lighting and reflectance. In Section 3 we derive lower bounds on the accuracy of spherical harmonic representations for objects of arbitrary shape and albedos under infinitely distant lighting. Finally, in Section 4 we examine the case of light sources at a finite distance from an object. Proofs are omitted for lack of space and will appear in a technical report.

2 Overview: Approximation by Spherical Harmonics

Basri and Jacobs [1] and Ramamoorthi and Hanrahan [14] constructed an analytically derived representation of the images produced by a convex, Lambertian object illuminated by distant light sources. Below we provide a brief outline of their results.

According to Lambert's law [9], which states that matt materials reflect light uniformly in all directions, a surface point with normal \mathbf{n} and albedo ρ illuminated by light arriving in direction \mathbf{l} and intensity i reflects light according to the following equation:

$$E_{\text{direc}} = \max(0, \langle \rho \mathbf{n}, i \mathbf{l} \rangle) , \quad (1)$$

where $\langle \cdot, \cdot \rangle$ denotes the usual inner product between vectors.

If we now consider a collection of directional (point) light sources placed at infinity, we can express the intensity of lighting as a non-negative function on the unit sphere $i(\mathbf{l})$. We can then express $i(\mathbf{l})$ as a sum of spherical harmonics (similar to a Fourier basis for R^n). We denote spherical harmonics by Y_{nm} ($n = 0, 1, 2, \dots$; $m = -n, \dots, n$). Then, $i(\mathbf{l}) = \sum_{n=0}^{\infty} \sum_{m=-n}^n \ell_{nm} Y_{nm}(\mathbf{l})$. Reflectance is then obtained from lighting by a convolution on the surface of a sphere, and using the Funk-Hecke theorem (see, e.g., [5]) the intensity of a point in an image E is given by:

$$E = \sum_{n=0}^{\infty} \sum_{m=-n}^n \alpha_n \ell_{nm} (\rho Y_{nm}(\mathbf{n})) , \quad (2)$$

with $\alpha_n = \pi, 2\pi/3, \pi/4, \dots$. For the specific case of a single directional source of intensity i and direction \mathbf{l} we have $\ell_{nm} = i Y_{nm}(\mathbf{l})$, and the harmonic expansion becomes

$$E_{\text{direc}} = \sum_{n=0}^{\infty} \sum_{m=-n}^n \alpha_n (i Y_{nm}(\mathbf{l})) (\rho Y_{nm}(\mathbf{n})) , \quad (3)$$

The coefficients tend to zero as $O(n^{-2.5})$ when $n \rightarrow \infty$. For this reason we can limit ourselves to just a few low order harmonics:

$$E \approx E^N = \sum_{n=0}^N \sum_{m=-n}^n \alpha_n \ell_{nm} (\rho Y_{nm}(\mathbf{n})) , \quad (4)$$

and evaluate the quality of the approximation using the *relative squared error*, defined as:

$$\varepsilon = \frac{\|E - E^N\|^2}{\|E\|^2} , \quad (5)$$

with the norm $\|f\|^2$ of a function f is defined as the integral of f^2 over its entire domain.

The quality of this approximation depends on the frequencies present in the lighting function $i(\mathbf{l})$. Consider the reflectance of a sphere with uniform

albedo (the so called *reflectance function*). If the sphere is illuminated by a single directional source then the approximation error is given by:

$$\varepsilon_{1,sph} = \frac{\sum_{n=N}^{\infty} (2n+1)\alpha_n^2}{\sum_{n=0}^{\infty} (2n+1)\alpha_n^2} . \quad (6)$$

(This is obtained from (5) using the orthonormality of Y_{nm} .) In particular, for orders 0, 1 and 2 the relative squared error is, respectively, 65.5%, 12.5% and 0.78% (corresponding to accuracy of 37.5%, 87.5% and 99.22%). The approximation is better if lighting contains mainly low frequency components and is worse if lighting includes many high frequencies, but even with high frequencies the error is bounded [1].

Reflectance functions, however, capture only general properties of images of Lambertian objects under specific lighting and ignore properties of a particular object. In particular, objects differ in shape and color, giving rise to different distributions of normals and albedos. In addition, foreshortening distorts the distribution of normals, and, due to occlusion, only the normals facing the camera are visible in an image. The accuracy obtained by low order harmonic approximations for images of objects change as well. In particular, there exist lighting conditions under which low order harmonic approximations of the images of objects are arbitrarily bad. However, we will show in the next section that on average the low order harmonic approximations for images of objects of *arbitrary* shape and albedo are accurate, and that the accuracies derived for reflectance functions provide in fact lower bounds to the average accuracies for any convex object.

3 Infinitely Distant Light

3.1 Basic Case: Uniformly Distributed Point Light Source

Lambert's law maintains a useful duality property. The formula (1) describing the light reflected by a point due to a directional source is symmetric, so that exchanging albedo ρ by light intensity i or normal \mathbf{n} by light direction \mathbf{l} maintains the same amount of light reflected. This duality relation is maintained also if we consider a discrete set of K surface points and a discrete set of J directional sources. In addition, it is maintained by every term of the spherical harmonic representation (4) and consequently by the expression for the relative squared error (5). Below we use this duality property to prove an error bound for arbitrary object approximation.

The approximation error for some light configurations can be arbitrarily large [1], so we cannot hope to bound the error for any light. We can try instead to describe a typical case by averaging the error over different illumination conditions. We consider first the case of a single directional light source. Since we have no prior information about the direction of this source, we assume that it is

drawn with a uniform distribution over a sphere around the object. For now we assume that the intensity of the light source is fixed and relax this assumption later.

To compute the error for this case we can consider the dual problem. Consider a single point on the surface of the object under all directional lightings. If we exchange object for light we will obtain a sphere with uniform albedo illuminated by a directional light source with arbitrary direction and intensity. The approximation error then is given by (6) and is independent of both the direction of the light source or its intensity. This argument can be applied to every point on the surface of the object. For every such point we obtain by the dual formulation a sphere illuminated by a different directional source, but the accuracy of the approximation remains the same. Consequently, the average error too is given by (6). This implies that for an arbitrary object the approximation error is constant depending only on the approximation order, and is independent on the object geometry and albedos.

We can readily extend this argument also to single directional sources with intensities drawn from an arbitrary distribution. All we need to assume is that the distribution of intensity is independent of direction.

Conclusion: **if** a convex Lambertian object is illuminated by a single directional light source, uniformly distributed over the sphere, **then** the accuracy of the spherical harmonic approximation does not depend on the object geometry and albedo.

We verified this conclusion on several examples. Note that in the analysis we use harmonic decompositions in which the coefficients are determined so as to optimally fit the reflectance of a uniformly sampled sphere. These may not be the optimal coefficients for an object. Thus, these derivations only give *lower bounds* on the accuracy of low order harmonic approximations.

Table 1 shows the accuracy obtained by simulations with a 4D (first order) and 9D (second order) harmonic approximation for various objects illuminated by single directional sources. In each case we used random albedos (uniform albedos lead to similar accuracies). In the case of the face we estimated albedo by averaging 15 images of the face. Note that both the actual harmonic approximations and SVD provide very similar accuracies, indicating that spherical harmonics indeed form an optimal basis. In addition, mainly due to foreshortening, those accuracies are slightly higher than the bound (about 99.5% compared to 99.22% in the 9D case and 94-98% compared to 87.5% in the 4D case).

3.2 Multiple Light Sources

In practice objects are often illuminated with multiple light sources. Does the result we obtained for a single light source hold for more general illumination configurations? We address this question below.

We consider the case of lighting consisting of multiple sources with (possibly) different intensities. We present an expression for the approximation error

Table 1. Approximation accuracy obtained for images of various objects illuminated by single directional sources. For each case we show the bound computed numerically (SHB – spherical harmonic bound), the actual harmonic approximation with optimal coefficients obtained using least squares (LSH – least squares harmonics), and the best low dimensional approximation obtained with SVD.

4D approximation				9D approximation			
Object	SHB	LSH	SVD	Object	SHB	LSH	SVD
Sphere	87.47	87.57	87.60	Sphere	99.21	99.22	99.22
Hemisphere	87.54	95.56	95.73	Hemisphere	99.22	99.45	99.47
Random	87.53	94.00	94.31	Random	99.22	99.45	99.47
Real face	87.58	97.73	98.00	Real face	99.23	99.66	99.70

depending on the number of light sources and their intensities, and give a meaningful analysis of this dependence. As before, consider an object of arbitrary shape and albedo and assume that the direction of the light sources are drawn from a uniform distribution and that the intensities are drawn from a distribution that is independent of direction.

Let us consider illumination consisting of K point light sources with intensities i_k and directions \mathbf{l}_k : $i(\mathbf{l}) = \sum_{k=1}^K i_k \delta_{\mathbf{l}_k}$. We assume that the directions of these point sources are distributed independently and uniformly over the sphere (while their intensities are fixed). We denote the relative approximation error of order N (defined as in (5)) by $\varepsilon_{K, sph}$.

Evaluating this error we obtain the following expression:

$$\varepsilon_{K, sph} = \frac{\varepsilon_{1, sph}}{1 + V}, \quad (7)$$

where V is determined by the light intensities:

$$V = \frac{3}{8} \left(\left(\sum_{k=1}^K i_k \right)^2 / \sum_{k=1}^K i_k^2 - 1 \right) = \frac{3}{8} \left(\frac{\|\mathbf{i}\|_{l_1}^2}{\|\mathbf{i}\|_{l_2}^2} - 1 \right), \quad (8)$$

with $\|\mathbf{i}\|_{l_1}$ and $\|\mathbf{i}\|_{l_2}$ respectively denote the l_1 and l_2 norms of the vector \mathbf{i} of light intensities.

V can be interpreted as a measure of non-uniformity of light intensities. It is always non-negative, and equal to zero for a single directional source. V is largest (and the error is smallest) when all the intensities i_k are equal (for a fixed number of sources K). In this case $V = \frac{3}{8}(K - 1)$ and $V \rightarrow \infty$ when the number of light sources K tends to infinity. Here we obtain in the limit the uniform ambient light and, not surprisingly, the approximation error (7) becomes zero.

Conclusion: **if** we consider an arbitrary number of (uniformly distributed, independent) multiple directional sources, **then** the accuracy of a low order harmonic approximation for a convex Lambertian object of arbitrary shape and albedos is not less than with a single light source. In other words, a single light source is the worst case illumination for the spherical harmonic approximation.

3.3 Light From Above

Our results thus far were obtained under the assumption that directional light sources are distributed uniformly over the entire sphere (the same assumption is adopted in [11,12,13]). But in reality we often meet the situation that light is coming mainly from above. To incorporate this prior knowledge we substitute the operation of averaging over the sphere for averaging over the upper hemisphere. We derive a bound for this case that is not constant for every order, but depends on the object normals and albedos.

The formula we derive allows to compute the average approximation error for any object illuminated by a random directional light source on the hemisphere and to analyze how object geometry and albedos influence this error. Our analysis shows that, unlike in the previous case, there exist objects for which the average harmonic approximation is arbitrarily bad. However, in a typical experimental setup (horizontally oriented camera) due to foreshortening the error is typically almost the same as with light distributed over the entire sphere.

Consider an arbitrary object illuminated by a single directional source \mathbf{l} with intensity i , which is uniformly distributed over the upper hemisphere. Using the harmonic expansion (3) of the image, for every surface normal and light the approximation error for $N \geq 2$ is given by:

$$\varepsilon_{1,hs} = \frac{\varepsilon_{1,sph}}{1 + \overline{F}}. \quad (9)$$

Here \overline{F} is a mean value of a function F (defined below), which depends on the normals and albedos of the object: $\overline{F} = \int_{object} \rho F(\theta) / \int_{object} \rho$, where θ is the angle between a surface normal and the vertical direction (θ varies from 0 to π). $F(\theta)$ is given by

$$F(\theta) = \frac{\sqrt{3}}{\pi} \sum_{n=0, n \neq 1}^{\infty} \alpha_n^2 \sqrt{2n+1} (Y_{n0}(\theta)Y_{10}(\theta) + \sqrt{2n(n+1)}Y_{n1}(\theta)Y_{11}(\theta)) \quad (10)$$

One can see the dependence of F (Figure 1) on the direction of the normals of the object.

Let us now analyze the expression (9). A positive value of \overline{F} reduces the error relative to the case of light distributed over the entire sphere. A negative value of \overline{F} increases this error (in the worst case to an arbitrarily large value). If most of the object normals (taking into account albedos) are directed upward then $\varepsilon_{1,hs}$ is smaller than the error for the sphere $\varepsilon_{1,sph}$. And vice versa, the more normals look downward, the greater is the error.

Foreshortening also affects this error, as it affects the density of the sampled normals. A typical setting is when light comes from above (sunlight or indoor ceiling lights) and the camera's optical axis is horizontal. In this case we average over F scaled by $\sin \theta$. It appears that in this case the effect of foreshortening "helps" us: normals that spoil the approximation the most (directed toward the ground) occur less frequently due to foreshortening, while normals directed to

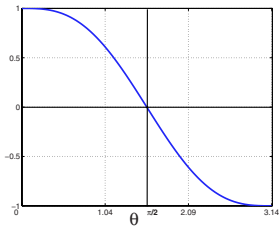


Fig. 1. The function $F(\theta)$. F is positive for $0 \leq \theta \leq \pi/2$ leading to improved accuracies for normals facing upward and negative otherwise, leading to decreased accuracies for normals facing downward.

the camera usually dominate the image. As a consequence the accuracy obtained with low order harmonic approximations is close to $1 - \varepsilon_{1,sph}$ ($F(\theta)$ is close to zero, see Figure 1).

Conclusion: **if** light distribution prior is non-uniform (from above in our analysis), **then** the accuracy of the spherical harmonic approximation depends on the object geometry. An object whose normals face the light (i.e. the dominant light direction, upwards in our analysis) is better approximated by spherical harmonics, than an object whose normals face the “dark side”.
In the typical setting that light is coming from above and the optical axis is horizontal, the approximation accuracy is usually close to the accuracy in the case of uniformly distributed light $1 - \varepsilon_{1,sph}$.

Table 2 shows the accuracy of the first five approximation orders for real and simulated objects. One can compare the results for two light distributions: when light source is distributed over the hemisphere (the first three rows) and when light source is distributed over the sphere (the bottom row).

Table 2. The first three rows show the n -th order approximation accuracy for different objects when light is coming from above. “Basic accuracy” refers to the accuracy when the light is distributed uniformly over the entire sphere.

Object	0	1	2	3	4	5	\overline{F}
Jurassic	26.99	85.40	99.09	99.09	99.77	99.77	-0.1440
Dinosaur	36.47	87.47	99.21	99.21	99.80	99.80	-0.0280
Face	58.86	91.77	99.49	99.49	99.87	99.87	0.5193
Basic accuracy	37.50	87.50	99.22	99.22	99.80	99.80	–

4 Near Light

Computer vision studies of lighting largely make the assumption of infinitely distant light. But in many images light is coming from nearby sources or is reflected from nearby objects, as for example in images taken in an indoor environment. It is then important to relax this assumption, or at least to determine what distance is sufficient to be considered infinite. In this section we attack this problem theoretically by analyzing a simplified near light model and practically by conducting experiments with near lighting.

Handling near light is complex because of two main problems. First, light originating from the same light source approaches different points on the object from different directions. Second, we can no longer assume that the intensity arriving at each point on the object is constant since it decreases as the squared distance to the light source. These problems imply in particular that we have to take into account the position of light sources and not only their direction, and thus lighting is no longer a function defined on the surface of a sphere. However, we will show an approach to still uses spherical harmonics. As before, our analysis accounts for attached shadows and will ignore the effect of cast shadows. Notice that in the case of near light the extent of attached shadows vary according to the distance to the light source.

4.1 Harmonic Approximation with Light at Finite Distance

Here we present an analytical model of the approximation of the reflectance function by spherical harmonics taking into account light at a finite distance. The model we describe assumes that all light sources are placed in some fixed distance from the object. The full treatment of near lighting requires taking into account images obtained by illuminating an object by collections of light sources at different distances from an object.

Consider a sphere of radius r with unit albedo centered at the origin (see Figure 2). Let \mathbf{p} denote a point on the sphere, and let $\mathbf{n}(\mathbf{p})$ denote the surface normal at \mathbf{p} . (Note that $\mathbf{p} = r\mathbf{n}$.) Assume that the sphere is illuminated by a point light source drawn from a uniform distribution over the sphere of radius $r + R$ also centered at the origin. Denote by \mathbf{u} a unit vector pointing toward the light source (so the light source is positioned at $(r + R)\mathbf{u}$), and let i denotes the light source intensity.

Denote by \mathbf{l} the vector from \mathbf{p} to the light source, namely, $\mathbf{l} = (r + R)\mathbf{u} - \mathbf{p}$, so that \mathbf{l} is the incident direction of the source at \mathbf{p} . The intensity that arrives at \mathbf{p} due to this source then is given by $i/\|\mathbf{l}\|^2$. Using Lambert's law the reflected intensity at \mathbf{p} due to this incident light is given by $E_{point} = \frac{i}{\|\mathbf{l}\|^2} \max(\langle \mathbf{n}, \mathbf{l} \rangle, 0)$.

Now consider a lighting function $i(\mathbf{u})$ describing the intensity of the light emitted from any number of sources placed on the sphere of radius $r + R$, the light reflected by the sphere of radius r due to these sources can be written as a convolution on the surface of a sphere of the lighting function $i(\mathbf{u})$ with the kernel

$$k = \max\left(\frac{\langle \mathbf{n}, \mathbf{l} \rangle}{\|\mathbf{l}\|^2}, 0\right). \quad (11)$$

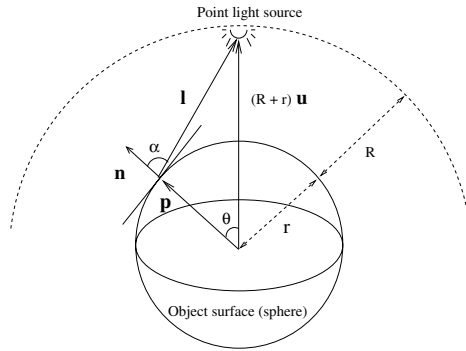


Fig. 2. A sphere of radius r is illuminated by a point light source. R is the distance between the sphere and the light. p denotes a point on the sphere and n denotes its normal vector. l is the incident lighting vector, and u is a unit vector directed toward the light source.

Expressing k as a function of θ , where θ is the angle between the light source vector h and the surface normal n we obtain:

$$k(\theta) = \max \left(\frac{(r + R) \cos(\theta) - r}{(R^2 + 2r(r + R)(1 - \cos(\theta)))^{3/2}}, 0 \right). \quad (12)$$

(Notice that k is a valid convolution kernel since it is rotationally symmetric about the north pole.) The harmonic expansion of $k(\theta)$ can be derived using the coefficients $k_n = 2\pi \int_0^\pi k(\theta) Y_{n0}(\theta) \sin(\theta) d\theta$. Integration can be limited to the positive portion of $k(\theta)$, so integration limits will now depend on the distances R and r . It is worth noting that unlike the case of infinitely distant light, harmonics of the odd orders are not eliminated by the kernel.

The relative squared energy concentrated in the first N harmonics is expressed by $\left(\sum_{n=0}^N k_n^2 / \sum_{n=0}^{\infty} k_n^2 \right)$. We can compute this relation for every finite N : the numerator is evaluated numerically and for the denominator we have the explicit formula:

$$\sum_{n=0}^{\infty} k_n^2 = \frac{\pi}{r^2 R^2} \left(\frac{2r(r + R) - 4rR + R^2 \log(1 + 2\frac{r}{R})}{4r(r + R)} \right). \quad (13)$$

4.2 Conclusions from the Simple Model

Figure 3 presents the dependence of the approximation accuracy on the distance to the light source. The distance is relative to the object size (R/r in the notation of our model). Each graph represents one approximation order (from zero to third). One can see that for extremely near light approximation accuracy is close to zero, but grows rapidly as the distance to the light increases approaching the accuracy values for infinitely distant light.

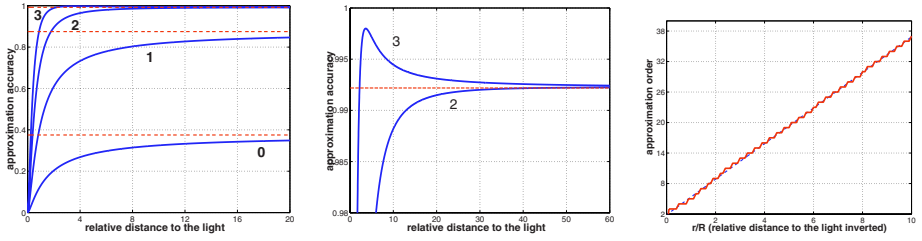


Fig. 3. Left: approximation accuracy as a function of the distance to the light source for approximation orders 0-3 (solid graphs). Dashed lines: asymptotes. Middle: a zoom-in version of the upper left portion of the left plot. Right: Order of approximation required to reach a 99.22% accuracy as a function of r/R .

The accuracy grows faster for higher approximation orders. This implies that for higher orders the light behaves as "infinitely distant" already at smaller distances. As an example consider the second and third orders (Figure 3(middle)). They both have the limit value of 99.22%. (We know from distant light analysis [1,14] that odd approximation orders higher than 1 do not contribute energy.) While the second order accuracy exceeds 99% from distance 13, the third order reaches this value already from the distance 2. Therefore for near light a third order approximation can be considerably more accurate than a second order.

As we increase the number of spherical harmonics we use, we can cope with arbitrarily close light. For example, to achieve an approximation accuracy of 99.22% we have the dependency shown in Figure 3(right). We see that the order of approximation we need is roughly inversely proportional to the distance of the light.

4.3 Experiments

We performed simulations on realistic objects to test the accuracy obtained with spherical harmonic approximations under near lighting.

We present results of simulations with a synthetic object model of a dinosaur's head ("Jurassic") obtained using the 3D Studio software. We used several light sources and moved them from very near positions to very far, rendering images for each light distance. Figure 4 shows the accuracies obtained for the first three approximation orders using the coefficients determined by our model and the coefficients obtained by least squares fitting.

We see that the approximation accuracy behaves similarly to the prediction of our model (apart from an undershoot for order zero). Starting with a small values for very near light, the accuracy for every order grows very fast and tends to its limit value for large distances. We see that even the distance scale in this simulation is very similar to the one obtained from the model.

We also performed near light experiments with two real objects: a human face and a dinosaur toy. 3D models of the objects were obtained with a laser scanner. Note that both objects are not exactly Lambertian, and some amount

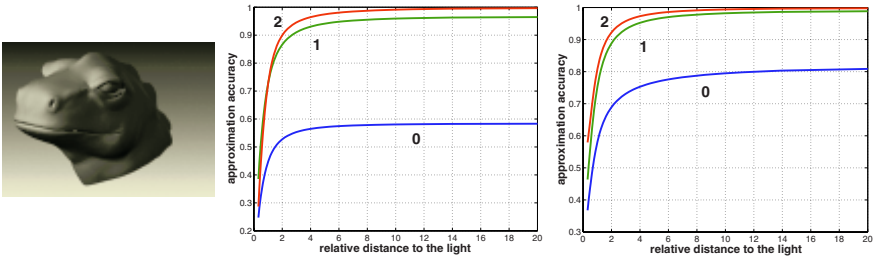




Fig. 4. Near light accuracy for Jurassic with harmonic approximations using the coefficients determined by our model (left) and the coefficients determined using least squares optimization (right).

Table 3. Least squares accuracy of harmonic approximations for a rendered model of a face (left) and a dinosaur (right). Approximate face radius is 15 cm, approximate dinosaur dimensions are: length 38 cm, width 8 cm, height 13 cm.



Distance	0	1	2	3
20 cm	77.61	91.14	92.20	92.47
50 cm	84.79	94.44	95.04	95.28
70 cm	89.56	95.00	95.40	95.63
100 cm	89.52	96.19	96.59	96.78
120 cm	92.72	97.07	97.28	97.40



Distance	0	1	2	3
20 cm	84.53	91.74	92.51	92.75
50 cm	86.49	92.82	93.49	93.68
80 cm	87.91	93.70	94.25	94.41
110 cm	88.08	93.30	93.90	94.03
140 cm	89.80	93.91	94.42	94.52

of cast shadows were present in the images. We tested the accuracies obtained for lighting at five distances using 15 pictures with varying lighting positions. The accuracies obtained are shown in Table 3. It can be seen that approximation accuracies are very high even for extremely near light.

Finally, we reconstructed the 3D shape of a dinosaur from 15 images obtained with lighting at 20 cm distance from the object using the photometric stereo method proposed in [2]. We used a first order (4D) harmonic approximation. This method uses factorization to recover lighting and shape fitting the obtained shape matrix to a spherical harmonic decomposition. The procedure allows the recovery of shape up to a 7 parameter scaled Lorentz ambiguity. To allow comparison of our results to ground truth we resolved this ambiguity manually. Figure 5 shows a subset of the images used for reconstruction, and Figure 6 shows the reconstruction obtained and for comparison the 3D shape obtained with a laser scanner. As can be seen although the object is illuminated by a very proximate lighting reconstruction is quite accurate. Similar reconstruction results were obtained with more distant lighting.



Fig. 5. Three out of 15 images with extremely near light (20 cm) used for photometric stereo reconstruction.

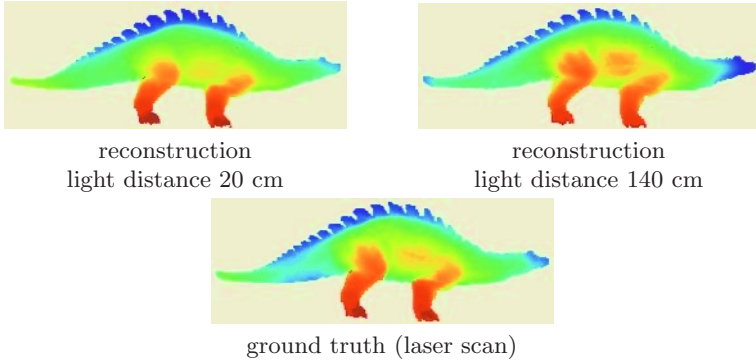


Fig. 6. Photometric stereo reconstruction results (depth maps). Colors (intensities) encode depth $z(x, y)$.

5 Summary

In this paper we have examined the use of spherical harmonic representations to model the images of Lambertian objects seen under arbitrary lighting, and extended its applicability to several cases of practical importance. We showed that under distant lighting, with reasonable assumptions on the distribution of light sources, the average accuracy of spherical harmonic representations can be bound from below independently of the shape and albedos of the object. We further examined the case of light coming from above and provided an analytic expression for the accuracy obtained in this case. Finally, we derived a model to compute the accuracy of low-dimensional harmonic representations under near lighting.

Our analysis demonstrates that spherical harmonic representations provide accurate modelling of lighting effects for a wide range of lighting conditions.

Acknowledgements. Research was supported in part by the European Community grant number IST-2000-26001 and by the Israel Science Foundation grants number 266/02. The vision group at the Weizmann Inst. is supported in part by the Moross Laboratory for Vision Research and Robotics.

References

1. R. Basri, D.W. Jacobs, "Lambertian reflectances and linear subspaces," *PAMI*, **25**(2), 2003.
2. R. Basri and D.W. Jacobs, "Photometric stereo with general, unknown lighting," *CVPR*, Vol. II: 374–381, 2001.
3. P. Bellhumeur, D. Kriegman. "What is the Set of Images of an Object Under All Possible Lighting Conditions?", *CVPR*: 270–277, 1996.
4. R. Epstein, P. Hallinan, A. Yuille. " 5 ± 2 Eigenimages Suffice: An Empirical Investigation of Low-Dimensional Lighting Models," *IEEE Workshop on Physics-Based Vision*: 108–116, 1995.
5. H. Groemer, *Geometric applications of Fourier series and spherical harmonics*, Cambridge University Press, 1996.
6. P. Hallinan. "A Low-Dimensional Representation of Human Faces for Arbitrary Lighting Conditions", *CVPR*: 995–999, 1994.
7. K. Hara, K. Nishino, K. Ikeuchi. "Determining Reflectance and Light Position from a Single Image Without Distant Illumination Assumption", *ICCV*: 560–567, 2003.
8. M. Kirby, L. Sirovich, "The application of the Karhunen-Loeve procedure for the characterization of human faces", *PAMI*, **12**(1): 103–108, 1990.
9. J. Lambert, "Photometria Sive de Mensura et Gradibus Luminis, Colorum et Umbrae," Eberhard Klett, 1760.
10. H. Murase, S. Nayar. Visual learning and recognition of 3D objects from appearance. *International Journal of Computer Vision*, **14**(1): 5–25, 1995.
11. P. Nillius, J.-O. Eklundh. "Low-Dimensional Representations of Shaded Surfaces under Varying Illumination", *CVPR*: 185–192, 2003.
12. P. Nillius, J.-O. Eklundh. "Phenomenological Eigenfunctions for Image Irradiance", *ICCV*: 568–575, 2003.
13. R. Ramamoorthi, "Analytic PCA construction for theoretical analysis of lighting in a single image of a Lambertian object," *PAMI*, **24**(10): 1322–1333, 2002.
14. R. Ramamoorthi, P. Hanrahan, "On the relationship between radiance and irradiance: determining the illumination from images of convex Lambertian object." *JOSA*, **18**(10): 2448–2459, 2001.
15. R. Ramamoorthi, P. Hanrahan, "An efficient representation for irradiance environment maps," *Siggraph*: 497–500, 2001.
16. D. Simakov, D. Frolova, R. Basri, "Dense shape reconstruction of a moving object under arbitrary, unknown lighting," *ICCV*: 1202–1209, 2003.
17. M. Turk, A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, **3**(1): 71–96, 1991.
18. A. Yuille, D. Snow, R. Epstein, P. Bellhumeur, "Determining Generative Models of Objects Under Varying Illumination: Shape and Albedo from Multiple Images Using SVD and Integrability", *IJCV*, **35**(3): 203–222, 1999.

Characterization of Human Faces under Illumination Variations Using Rank, Integrability, and Symmetry Constraints^{*}

S. Kevin Zhou, Rama Chellappa, and David W. Jacobs

Center for Automation Research
University of Maryland, College Park MD 20742, USA

Abstract. Photometric stereo algorithms use a Lambertian reflectance model with a varying albedo field and involve the appearances of only one object. This paper extends photometric stereo algorithms to handle all the appearances of all the objects in a class, in particular the class of human faces. Similarity among all facial appearances motivates a rank constraint on the albedos and surface normals in the class. This leads to a factorization of an observation matrix that consists of exemplar images of different objects under different illuminations, which is beyond what can be analyzed using bilinear analysis. Bilinear analysis requires exemplar images of different objects under same illuminations. To fully recover the *class-specific* albedos and surface normals, integrability and face symmetry constraints are employed. The proposed linear algorithm takes into account the effects of the varying albedo field by approximating the integrability terms using only the surface normals. As an application, face recognition under illumination variation is presented. The rank constraint enables an algorithm to separate the illumination source from the observed appearance and keep the illuminant-invariant information that is appropriate for recognition. Good recognition results have been obtained using the PIE dataset.

1 Introduction

Recovery of albedos and surface normals has been studied in the computer vision community for a long time. Usually a Lambertian reflectance model, ignoring both attached and cast shadows, is employed. Early works from the shape from shading (SFS) literature assume a constant albedo field: this assumption is not valid for many real objects and thus limits the practical applicability of the SFS algorithms. Early photometric stereo approaches require knowledge of lighting conditions, but full control of the lighting sources is also constraining. Recent research efforts [1,2,4,5,6,7,8] attempt to go beyond these restrictions by (i) using a varying albedo field, a more accurate model of the real world, and (ii) assuming no prior knowledge or requiring no control of the lighting sources. As a consequence, the complexity of the problem has also increased significantly.

^{*} Partially supported by the DARPA/ONR Grant N00014-03-1-0520.

If we fix the imaging geometry and only move the lighting source to illuminate one object, the observed images (ignoring the cast and attached shadows) lie in a subspace completely determined by three images illuminated by three independent lighting sources [4]. If an ambient component is added [6], this subspace becomes 4-D. If attached shadows are considered as in [1,2], the subspace dimension grows to infinity but most of its energy is packed in a limited number of harmonic components, thereby leading to a low-dimensional subspace approximation. However, all the photometric-stereo-type approaches (except [5]) commonly restrict themselves to using *object-specific* samples and cannot handle the appearances not belonging to the object. In this paper, we extend the photometric stereo algorithms to handle all the appearances of all the objects in a class, in particular the human face class.

To this end, we impose a rank constraint (i.e. a linear generalization) on the albedos and surface normals of all human faces. This rank constraint enables us to accomplish a factorization of the observation matrix that decomposes a *class-specific* ensemble into a product of two matrices: one encoding the albedos and surfaces normals for a class of objects and the other encoding blending linear coefficients and lighting conditions. A class-specific ensemble consists of exemplar images of different objects under different illuminations, which can not be analyzed using bilinear analysis [14]. Bilinear analysis requires exemplar images of different objects under the same illuminations. Because a factorization is always up to an invertible matrix, a full recovery of the albedos and surface normals is not a trivial task and requires additional constraints. The surface integrability constraint [9,10] has been used in several approaches [6,8] to successfully perform the recovery task. The symmetry constraint has also been employed in [7,11] for face images. In Section 3, we present an approach which fuses these constraints to recover the albedos and surface normals for the face class, even in the presence of shadows. More importantly, this approach takes into account the effects of the varying albedo field by approximating the integrability terms using only the surface normals instead of the product of the albedos and the surface normals. Due to the nonlinearity embedded in the integrability terms, regular algorithms such as steepest descent are inefficient. We derive a linearized algorithm to find the solution.

In addition, the blending linear coefficients offer an illuminant-invariant identity signature, which is appropriate for face recognition under illumination variation. In Section 4, we first present a method for computing such coefficients and then report face recognition results using the PIE database [12].

1.1 Notations

In general, we denote a scalar by a , a vector by \mathbf{a} , and a matrix with r rows and c columns by $\mathbf{A}_{r \times c}$. The matrix transpose is denoted by \mathbf{A}^T , the pseudo-inverse by \mathbf{A}^\dagger . The matrix L_2 -norm is denoted by $\|\cdot\|_2$.

The following notations are introduced for the sake of notational conciseness and emphasis of special structure.

- Concatenation notations: \Rightarrow and \Downarrow .

\Rightarrow and \Downarrow mean horizontal and vertical concatenations, respectively. For example, we can represent a $n \times 1$ vector $\mathbf{a}_{n \times 1}$ by $\mathbf{a} = [a_1, a_2, \dots, a_n]^T = [\Downarrow_{i=1}^n a_i]$ and its transpose by $\mathbf{a}^T = [a_1, a_2, \dots, a_n] = [\Rightarrow_{i=1}^n a_i]$. We can use \Rightarrow and \Downarrow to concatenate matrices to form a new matrix. For instance, given a collection of matrices $\{\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n\}$ of size $r \times c$, we construct a $r \times cn$ matrix¹ $[\Rightarrow_{i=1}^n \mathbf{A}_i] = [\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n]$ and a $rn \times c$ matrix $[\Downarrow_{i=1}^n \mathbf{A}_i] = [\mathbf{A}_1^T, \mathbf{A}_2^T, \dots, \mathbf{A}_n^T]^T$. In addition, we can combine \Rightarrow and \Downarrow to achieve a concise notation. Rather than representing a matrix $\mathbf{A}_{r \times c}$ as $[a_{ij}]$, we represent it as $\mathbf{A}_{r \times c} = [\Downarrow_{i=1}^r [\Rightarrow_{j=1}^c a_{ij}]] = [\Rightarrow_{j=1}^c [\Downarrow_{i=1}^r a_{ij}]]$. Also we can easily construct ‘big’ matrices using ‘small’ matrices $\{\mathbf{A}_{11}, \mathbf{A}_{12}, \dots, \mathbf{A}_{1n}, \dots, \mathbf{A}_{mn}\}$ of size $r \times c$. The matrix $[\Downarrow_{i=1}^m [\Rightarrow_{j=1}^n \mathbf{A}_{ij}]]$ is of size $rm \times cn$, the matrix $[\Rightarrow_{i=1}^m [\Rightarrow_{j=1}^n \mathbf{A}_{ij}]]$ of size $r \times cmn$.

- Kronecker (tensor) product: \otimes .

It is defined as $\mathbf{A}_{m \times n} \otimes \mathbf{B}_{r \times c} = [\Downarrow_{i=1}^m [\Rightarrow_{j=1}^n a_{ij} \mathbf{B}]]_{mr \times nc}$.

2 Setting and Constraints

We assume a Lambertian imaging model with a varying albedo field and no shadows. A pixel h is represented as

$$h = p \mathbf{n}^T \mathbf{s} = \mathbf{t}^T \mathbf{s}, \quad (1)$$

where p is the albedo at the pixel, $\mathbf{n} \doteq [\hat{a}, \hat{b}, \hat{c}]^T$ is the unit surface normal vector at the pixel, $\mathbf{t}_{3 \times 1} \doteq [a \doteq p\hat{a}, b \doteq p\hat{b}, c \doteq p\hat{c}]^T$ is the product of albedo and surface normal, and \mathbf{s} specifies a distant illuminant (a 3×1 unit vector multiplied by the light’s intensity).

For an image \mathbf{h} , a collection of d pixels $\{h_i, i = 1, \dots, d\}$ ², by stacking all the pixels into a column vector \mathbf{h} , we have

$$\mathbf{h}_{d \times 1} \doteq [\Downarrow_i h_i] = [\Downarrow_i \mathbf{t}_i^T] \mathbf{s} = \mathbf{T}_{d \times 3} \mathbf{s}_{3 \times 1}, \quad (2)$$

where $\mathbf{T} \doteq [\Downarrow_i \mathbf{t}_i^T]$ contains all albedo and surface normal information about the object. We call the \mathbf{T} matrix the *object-specific albedo-shape* matrix.

In the case of photometric stereo, we have n images of the *same* object, say $\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n\}$, observed at a fixed pose illuminated by n different lighting sources, forming an *object-specific* ensemble. Simple algebraic manipulation gives:

$$\mathbf{H}_{d \times n} \doteq [\Rightarrow_i \mathbf{h}_i] = \mathbf{T} [\Rightarrow_i \mathbf{s}_i] = \mathbf{T}_{d \times 3} \mathbf{S}_{3 \times n}, \quad (3)$$

¹ We do not need the size of $\{\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n\}$ to be exactly same. We use matrices of the same size in this example for simplicity. For example, for $[\Rightarrow_{i=1}^n \mathbf{A}_i]$, we only need the number of rows of these matrices to be same.

² The index i corresponds to a spatial position $\mathbf{x} \doteq (x, y)$. If no confusion, we will interchange both notations. For instance, we might also use $\mathbf{x} = 1, \dots, d$.

where \mathbf{H} is the *observation matrix* and $\mathbf{S} \doteq [\Rightarrow_i \mathbf{s}_i]$ encodes the information about the illuminants. Hence photometric stereo is rank-3 constrained. Therefore, given at least three exemplar images for one object under three different independent illuminations, we can determine the identity of a new probe image by checking if it lies in the linear span of the three exemplar images [4]. This requires obtaining at least three images for each object in the gallery set, which may not be possible in many applications. Note that in this recognition setting, there is no need for the training set that is defined below; in other words, the training set is equivalent to the gallery set.

We follow [13] in defining a typical recognition protocol for face recognition algorithms. Three sets are needed: Gallery, probe, and training sets. The gallery set consists of images with known identities. The probe set consists of images whose identities are to be determined by matching with the gallery set. In addition, the training set is provided for the recognition algorithm to learn characteristic features of the face images. In general, we assume no identity overlap between the gallery set and the training set and often store only one exemplar image for each object in the gallery set. However, the training set can have more than one image for each object. In order to generalize from the training set to the gallery and probe sets, we note that all images in the training, gallery, and probe sets belong to the same face class, which naturally leads to a rank constraint.

2.1 The Rank Constraint

We impose the rank constraint on the \mathbf{T} matrix by assuming that any \mathbf{T} matrix is a linear combination of some basis matrices $\{\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_m\}$, i.e., there exist coefficients $\{f_j; j = 1, \dots, m\}$ such that

$$\mathbf{T}_{d \times 3} = \sum_{j=1}^m f_j \mathbf{T}_j = [\Rightarrow_j \mathbf{T}_j](\mathbf{f} \otimes \mathbf{I}_3) = \mathbf{W}_{d \times 3m}(\mathbf{f}_{m \times 1} \otimes \mathbf{I}_3), \quad (4)$$

where $\mathbf{f} \doteq [\Downarrow_j f_j]$, $\mathbf{W} \doteq [\Rightarrow_j \mathbf{T}_j]$, and \mathbf{I}_n denotes an identity matrix of dimension $n \times n$. Since the \mathbf{W} matrix encodes all albedos and surface normals for a class of objects, we call it a *class-specific albedo-shape* matrix. Similar rank constraints are widely found in the literature; see for example [20,21].

Substitution of (4) into (2) yields

$$\mathbf{h}_{d \times 1} = \mathbf{T}\mathbf{s} = \mathbf{W}(\mathbf{f} \otimes \mathbf{I}_3)\mathbf{s} = \mathbf{W}(\mathbf{f} \otimes \mathbf{s}) = \mathbf{W}_{d \times 3m} \mathbf{k}_{3m \times 1}, \quad (5)$$

where $\mathbf{k} \doteq \mathbf{f} \otimes \mathbf{s}$. This leads to a two-factor bilinear analysis [14]. Recently, a multilinear analysis has been proposed in [20,22].

With the availability of n images $\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n\}$ for *different* objects, observed at a fixed pose illuminated by n different lighting sources, forming a *class-specific* ensemble, we have

$$\mathbf{H}_{d \times n} = [\Rightarrow_i \mathbf{h}_i] = \mathbf{W}[\Rightarrow_i (\mathbf{f}_i \otimes \mathbf{s}_i)] = \mathbf{W}[\Rightarrow_i \mathbf{k}_i] = \mathbf{W}_{d \times 3m} \mathbf{K}_{3m \times n}, \quad (6)$$

where $\mathbf{K} \doteq [\Rightarrow_i (\mathbf{f}_i \otimes \mathbf{s}_i)] = [\Rightarrow_i \mathbf{k}_i]$. It is a rank- $3m$ problem. Notice that \mathbf{K} takes a special form.

The rank constraint generalizes many approaches in the literature and is quite easily satisfied. If $m = 1$, this reduces to the case of photometric stereo; if the surface normal is fixed and the albedo field lies in a rank- m linear subspace, we have (4) satisfied too. Interestingly, the ‘Eigenface’ approach [15] is just a special case of our approach, but this is only for a fixed illumination source. Suppose that the illuminant vector is $\tilde{\mathbf{s}}$. (5) and (6) reduce to equations:

$$\mathbf{h}_{d \times 1} = \mathbf{W}(\mathbf{f} \otimes \tilde{\mathbf{s}}) = \tilde{\mathbf{W}}_{d \times m} \mathbf{f}_{m \times 1}; \quad \mathbf{H}_{d \times n} = [\Rightarrow_i \mathbf{h}_i] = \tilde{\mathbf{W}}[\Rightarrow_i \mathbf{f}_i] = \tilde{\mathbf{W}}_{d \times m} \mathbf{F}_{m \times n}, \quad (7)$$

where $\tilde{\mathbf{W}} \doteq [\Rightarrow_i \mathbf{T}_i \tilde{\mathbf{s}}]$. Therefore, our approach can be regarded as a generalized ‘Eigenface’ analysis able to handle illumination variation.

Our immediate goal is to estimate \mathbf{W} and \mathbf{K} from the observation matrix \mathbf{H} . The first step is to invoke an SVD factorization, say $\mathbf{H} = \mathbf{U} \mathbf{A} \mathbf{V}^T$, and retain the top $3m$ components as $\mathbf{H} = \mathbf{U}_{3m} \mathbf{A}_{3m} \mathbf{V}_{3m}^T = \hat{\mathbf{W}} \hat{\mathbf{K}}$, where $\hat{\mathbf{W}} = \mathbf{U}_{3m}$ and $\hat{\mathbf{K}} = \mathbf{A}_{3m} \mathbf{V}_{3m}^T$. Thus, we can recover \mathbf{W} and \mathbf{K} up to a $3m \times 3m$ invertible matrix \mathbf{R} with $\mathbf{W} = \hat{\mathbf{W}} \mathbf{R}$, $\mathbf{K} = \mathbf{R}^{-1} \hat{\mathbf{K}}$. Additional constraints are required to determine the \mathbf{R} matrix. We use the integrability and symmetry constraints, both related to \mathbf{W} . Moreover, $\mathbf{K} = [\Rightarrow_i (\mathbf{f}_i \otimes \mathbf{s}_i)]$.

2.2 The Integrability Constraint

One common constraint used in SFS research is the integrability of the surface [9,10,6,8]. Suppose that the surface function is $z = z(\mathbf{x})$ with $\mathbf{x} \doteq (x, y)$, we must have $\frac{\partial}{\partial x} \frac{\partial z}{\partial y} = \frac{\partial}{\partial y} \frac{\partial z}{\partial x}$. If given the unit surface normal vector $\mathbf{n}(\mathbf{x}) \doteq [\hat{a}(\mathbf{x}), \hat{b}(\mathbf{x}), \hat{c}(\mathbf{x})]^T$ at pixel \mathbf{x} , we have $\frac{\partial}{\partial x} \frac{\hat{b}(\mathbf{x})}{\hat{c}(\mathbf{x})} = \frac{\partial}{\partial y} \frac{\hat{a}(\mathbf{x})}{\hat{c}(\mathbf{x})}$. In other words, with $\alpha(\mathbf{x})$ defined as an integrability constraint term,

$$\alpha(\mathbf{x}) \doteq \hat{c}(\mathbf{x}) \frac{\partial \hat{b}(\mathbf{x})}{\partial x} - \hat{b}(\mathbf{x}) \frac{\partial \hat{c}(\mathbf{x})}{\partial x} + \hat{a}(\mathbf{x}) \frac{\partial \hat{c}(\mathbf{x})}{\partial y} - \hat{c}(\mathbf{x}) \frac{\partial \hat{a}(\mathbf{x})}{\partial y} = 0. \quad (8)$$

If instead are given the product of albedo and surface normal $\mathbf{t}(\mathbf{x}) \doteq [a(\mathbf{x}), b(\mathbf{x}), c(\mathbf{x})]^T$ with $a(\mathbf{x}) \doteq p(\mathbf{x}) \hat{a}(\mathbf{x})$, $b(\mathbf{x}) \doteq p(\mathbf{x}) \hat{b}(\mathbf{x})$, and $c(\mathbf{x}) \doteq p(\mathbf{x}) \hat{c}(\mathbf{x})$, (8) still holds with \hat{a} , \hat{b} , and \hat{c} replaced by a , b , and c , respectively. Practical algorithms approximate the partial derivatives by forward or backward differences or other differences that use an inherent smoothness assumption. Hence, the approximations based on $\mathbf{t}(\mathbf{x})$ are very rough especially at places where abrupt albedo variation exists (e.g. the boundaries of eyes, iris, eyebrow, etc) since the smoothness assumption is seriously violated. We should by all means use $\mathbf{n}(\mathbf{x})$ in order to remove this effect.

2.3 The Symmetry Constraint

For a face image in a frontal view, one natural constraint is its symmetry about the central y -axis as proposed in [7,11]:

$$p(x, y) = p(-x, y); \quad \hat{a}(x, y) = -\hat{a}(-x, y); \quad \hat{b}(x, y) = \hat{b}(-x, y); \quad \hat{c}(x, y) = \hat{c}(-x, y), \quad (9)$$

which is equivalent to, using $\mathbf{x} \doteq (x, y)$ and its symmetric point $\bar{\mathbf{x}} \doteq (-x, y)$,

$$a(\mathbf{x}) = -a(\bar{\mathbf{x}}); \quad b(\mathbf{x}) = b(\bar{\mathbf{x}}); \quad c(\mathbf{x}) = c(\bar{\mathbf{x}}). \quad (10)$$

If a face image is in a non-frontal view, such a symmetry still exists but the coordinate system should be modified to take into account the view change.

3 The Recovery of Albedos and Surface Normals

The recovery task is to find from the observation matrix \mathbf{H} the *class-specific albedo-shape* matrix \mathbf{W} (or equivalently \mathbf{R}), which satisfies both the integrability and symmetry constraints, as well as the matrices \mathbf{F} and \mathbf{S} . Denote $\mathbf{R} \doteq [\Rightarrow_{j=1}^m [r_{aj}, r_{bj}, r_{cj}]]$ and $\hat{\mathbf{W}} \doteq [\Downarrow_{\mathbf{x}=1}^d \hat{\mathbf{w}}_{(\mathbf{x})}^T]$. As $\mathbf{W} \doteq [\Downarrow_{\mathbf{x}=1}^d [\Rightarrow_{j=1}^m [a_j(\mathbf{x}), b_j(\mathbf{x}), c_j(\mathbf{x})]]] = \hat{\mathbf{W}}\mathbf{R}$, we have

$$a_j(\mathbf{x}) = \hat{\mathbf{w}}_{(\mathbf{x})}^T r_{aj}, \quad b_j(\mathbf{x}) = \hat{\mathbf{w}}_{(\mathbf{x})}^T r_{bj}, \quad c_j(\mathbf{x}) = \hat{\mathbf{w}}_{(\mathbf{x})}^T r_{cj}; \quad j = 1, \dots, m. \quad (11)$$

Practical systems must take into account attached and cast shadows as well as sensor noise. The existence of shadows in principle increases the rank to infinity. But, if we exclude shadowed pixels or set them as missing values, we still have rank 3. Performing a SVD with missing values is discussed in [3,16].

In view of the above circumstances, we formulate the following optimization problem: minimizing over \mathbf{R} , \mathbf{F} , and \mathbf{S} the cost function \mathcal{E} defined as

$$\begin{aligned} \mathcal{E}(\mathbf{R}, \mathbf{F}, \mathbf{S}) &= \frac{1}{2} \sum_{i=1}^n \sum_{\mathbf{x}=1}^d \mathbf{i}_i(\mathbf{x}) \{h_{i(\mathbf{x})} - \hat{\mathbf{w}}_{(\mathbf{x})}^T \mathbf{R}(\mathbf{f}_i \otimes \mathbf{s}_i)\}^2 \\ &\quad + \frac{\lambda_1}{2} \sum_{j=1}^m \sum_{\mathbf{x}=1}^d \{\alpha_{j(\mathbf{x})}\}^2 + \frac{\lambda_2}{2} \sum_{j=1}^m \sum_{\mathbf{x}=1}^d \{\beta_{j(\mathbf{x})}\}^2, \\ &= \mathcal{E}_0(\mathbf{R}, \mathbf{F}, \mathbf{S}) + \lambda_1 \mathcal{E}_1(\mathbf{R}) + \lambda_2 \mathcal{E}_2(\mathbf{R}), \end{aligned} \quad (12)$$

where $\mathbf{i}_i(\mathbf{x})$ is an indicator function describing whether the pixel \mathbf{x} of the image \mathbf{h}_i is in shadow, $\alpha_{j(\mathbf{x})}$ is the integrability constraint term based only on surface normals as defined in (8), and $\beta_{j(\mathbf{x})}$ is the symmetry constraint term given as

$$\beta_{j(\mathbf{x})}^2 = \{a_j(\mathbf{x}) + a_j(\bar{\mathbf{x}})\}^2 + \{b_j(\mathbf{x}) - b_j(\bar{\mathbf{x}})\}^2 + \{c_j(\mathbf{x}) - c_j(\bar{\mathbf{x}})\}^2; \quad j = 1, \dots, m. \quad (13)$$

One approach could be to directly minimize the cost function over \mathbf{W} , \mathbf{F} , and \mathbf{S} . This is in principle possible but numerically difficult as the number of unknowns depends on the image size, which can be quite large in practice.

As shown in [17], the surface normals can be recovered up to a generalized bas-relief (GBR) ambiguity. To resolve the GBR ambiguity, we normalize the matrix \mathbf{R} by keeping $\|\mathbf{R}\|_2 = 1$. Another ambiguity between \mathbf{f}_j and \mathbf{s}_j is a nonzero scale, which can be removed by normalizing \mathbf{f} : $\mathbf{f}_j^T \mathbf{1} = 1$, where $\mathbf{1}_{m \times 1}$ is a vector of 1's.

To summarize, we perform the following task:

$$\min_{\mathbf{R}, \mathbf{F}, \mathbf{S}} \mathcal{E}(\mathbf{R}, \mathbf{F}, \mathbf{S}) \text{ subject to } \|\mathbf{R}\|_2 = 1, \mathbf{F}^T \mathbf{1} = 1. \quad (14)$$

An iterative algorithm can be designed to solve (14). While solving for \mathbf{F} and \mathbf{S} with \mathbf{R} fixed is quite easy, solving for \mathbf{R} given \mathbf{F} and \mathbf{S} is very difficult because the integrability constraint terms require partial derivatives of the surface normals that are nonlinear in \mathbf{R} . Regular algorithms such as steepest descent are inefficient. One main contribution in this paper is that we propose a linearized algorithm to solve for \mathbf{R} . Appendix-I presents the details of the complete algorithm.

To demonstrate how the algorithm works, we design the following scenario with $m = 2$ so that the rank of interest is $2 \times 3 = 6$. To defeat the photometric stereo algorithm, which requires one object illuminated by at least three sources, and the bilinear analysis which requires two fixed objects illuminated by at least the same three lighting sources, we synthesize eight images by taking random linear combinations of two basis objects illuminated by eight different lighting sources. Fig. 1 displays the synthesized images and the recovered class-specific albedo-shape matrix, which clearly shows the two basis objects. Our algorithm converges within 100 iterations.

One notes that the special case $m = 1$ of our algorithm can be readily applied to photometric stereo (with the symmetry constraint removed) to robustly recover the albedos and surface normals for one object.

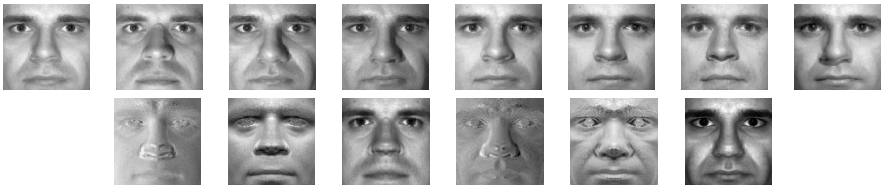


Fig. 1. Row 1: Eight Synthesized images that are random linear combinations of two basis objects illuminated by eight different lighting sources. Row 2: Recovered class-specific albedo-shape matrix showing the two basis objects (i.e. the three columns of \mathbf{T}_1 and \mathbf{T}_2).

4 Recognition Experiments

We study an extreme recognition setting with the following features: there is no identity overlap between the training set and the gallery and probe sets; only one image for one object is stored in the gallery set; the lighting conditions for the training, gallery and probe sets are completely unknown. The only known fact is that the face is in frontal view. Our strategy is to: (i) Learn \mathbf{W} from the

training set using the recovery algorithm described in Section 3; (ii) With W given, learn the identity signature f 's for both the gallery and probe sets using the recovery algorithm described in Section 4.1 and no knowledge of illumination directions; and (iii) Perform recognition using the nearest correlation coefficient. Suppose that one gallery image g has its signature f_g and one probe image p has its signature f_p , their correlation coefficient is $cc(p, g) = (f_p, f_g) / \sqrt{(f_p, f_p)(f_g, f_g)}$, where (x, y) is an inner-product such as $(x, y) = x^T \Sigma y$ with Σ learned or given.

4.1 Separating Illumination

With the class-specific albedo-shape matrix W available, we proceed to solve the problem of separating illumination, *v.i.z.*, for an arbitrary image h , find the illuminant vector s and the identity signature f . For convenience of recognition, we normalize f to the same range: $f^T \mathbf{1} = 1$. Appendix-II presents the recovery algorithm which infers the shadow pixels as well.

4.2 PIE Dataset

We use the Pose and Illumination and Expression (PIE) dataset [12] in our experiment. Fig. 2 shows the distribution of all 21 flashes used in PIE and their estimated positions using our algorithm. Since the flashes are almost symmetrically distributed about the head position, we only use 12 of them distributed on the right half of the unit sphere in Fig. 2. In total, we used $68 \times 12 = 816$ images in a fixed view as there are 68 subjects in the PIE database. Fig. 2 also displays one PIE object under the selected 12 illuminants. Registration is performed by aligning the eyes and mouth to canonical positions. No flow computation is carried out for further alignment. We use cropped face regions. After the pre-processing step, the actual image size is 50×50 . Also, we only study gray images by taking the average of the red, green, and blue channels. We use all 68 images under one illumination to form a gallery set and under another illumination to form a probe set. The training set is taken from sources other than the PIE dataset. Thus, we have 132 tests, with each test giving rise to a recognition score.

4.3 Recognition Performance

The training set is first taken from the Yale illumination database [8]. There are only 10 subjects (i.e. $m = 10$) in this database and each subject has 64 images in frontal view illuminated by 64 different lights. We only use 9 lights and Fig. 2 shows one Yale object under 9 lights.

Table 1 lists the recognition rate for all test protocols for the PIE database using the Yale database as the training set. Even with $m = 10$, we obtain quite good results. One observation is that when the flashes become separated, the recognition rate decreases. Also, using images under frontal or near-frontal illuminants as galleries produces good results. For comparison, we also implemented the ‘Eigenface’ approach [15] by training the projection directions from the same training set. Its average recognition rate is only 35% while ours is 67%.

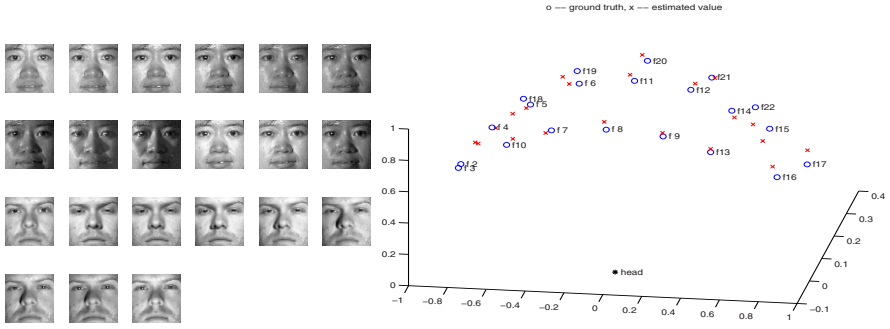


Fig. 2. Left: Rows 1-2 display one PIE object under the selected 12 illuminants and Rows 3-4 one Yale object under 9 lights used in the training set. Right: Flash distribution in the PIE database. For illustrative purposes, we move their positions on a unit sphere as only the illuminant directions matter. ‘o’ means the ground truth and ‘x’ the estimated values. It is quite accurate for estimation of directions of flashes near frontal pose. But when the flashes are very off-frontal, accuracy goes down slightly.

Table 1. Recognition rate obtained by our rank constrained approach using the Yale database (the left number in each cell) and Vetter database (the right number in each cell) as the training set. ‘F’ means 100 and ‘fnm’ flash number nm .

Gallery	f08	f09	f11	f12	f13	f14	f15	f16	f17	f20	f21	f22	Average
Probe													
f08	-	96/F	96/99	87/99	66/97	60/97	46/79	29/72	22/43	85/99	78/97	53/93	65/88
f09	94/F	-	96/99	96/99	90/99	87/99	56/97	40/91	24/60	84/97	96/97	68/97	75/94
f11	94/99	91/99	-	97/F	72/F	72/F	38/90	28/76	16/65	F/F	94/F	51/99	69/93
f12	88/99	94/99	97/F	-	88/F	93/F	57/F	41/93	28/76	94/F	F/F	76/F	78/97
f13	56/99	87/99	59/F	85/F	-	F/F	90/F	71/F	50/88	54/99	87/F	F/F	76/99
f14	51/99	85/99	63/F	93/F	F/F	-	90/F	66/F	49/96	59/99	91/F	99/F	77/99
f15	33/84	40/94	37/93	49/F	85/F	88/F	-	93/F	78/F	32/88	49/F	97/F	62/96
f16	19/69	26/87	26/78	32/90	59/F	44/F	84/F	-	93/F	26/69	31/F	63/F	46/89
f17	14/44	28/60	19/51	26/71	50/84	41/91	68/99	94/F	-	19/56	26/75	44/94	39/75
f20	90/97	85/97	99/F	97/F	65/F	69/F	38/90	26/74	21/68	-	93/F	53/F	67/93
f21	79/97	94/97	93/F	F/F	88/F	94/F	62/F	49/97	28/82	91/F	-	76/F	78/98
f22	43/90	65/97	46/96	75/F	99/F	99/F	97/F	76/F	59/99	43/97	74/F	-	70/98
Average	60/89	72/93	66/92	76/96	78/98	77/99	66/96	56/91	42/80	63/91	74/96	71/98	67/93

Generalization capacity with $m = 10$ is rather restrictive. We now increase m from 10 to 100 by using the Vetter’s 3D face database [18]. As this is a 3D database, we actually have W available. However, we believe that using a training set of $m = 100$ from other sources can yield similar performance. Table 1 presents the recognition rates. Significant improvements have been achieved due to the increase in m . The average rate is 93%. This seems to suggest that a moderate rank of 100 is enough to span the entire face space under a fixed view.

As a comparison, Romdhani et. al. [18] reported the recognition rates only with ‘f12’ being the gallery set and their average is 98% while ours is 96%. Our approach is very different from [18]. In [18] depths and texture maps of explicit 3D face models are used, while our approach is image-based and recovers albedos and surface normals. 3D models can be then be reconstructed. In the

experiments, (i) we use the ‘illum’ part of the PIE database that is close to the Lambertian model and they use the ‘light’ part that includes an ambient light; (ii) we use gray-valued images and they use color images; (iii) we assume known pose but unknown illumination but they assume unknown pose but known illumination; and (iv) compared to [18], our alignment is rather crude and can be improved using flow computations. We believe that our recognition rates can be boosted using the color images and finer alignment.

5 Conclusions

We presented an approach that naturally combines the rank-constraint for identity with illumination modeling. By using the integrability and symmetry constraints, we then achieved a linear algorithm that recovers the albedos and surface normals for a class of face images under the most general setting, i.e., the observation matrix consists of different objects under different illuminations. Further, after separating the illuminations, we obtained illumination-invariant identity signatures which produced good recognition performances under illumination variations. We still need to investigate pose variations and extreme lighting conditions that cause more shadows.

References

1. R. Basri and D. Jacobs, “Lambertian reflectance and linear subspaces,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218 - 233, 2003.
2. R. Basri and D. Jacobs, “Photometric stereo with general, unknown lighting,” *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 374-381, 2001.
3. D. Jacobs, “Linear fitting with missing data for structure-from-motion,” *Computer Vision and Image Understanding*, vol. 82, pp. 57 - 81, 2001.
4. A. Shashua, “On photometric issues in 3D visual recognition from a single 2D image,” *International Journal of Computer Vision*, vol. 21, no. 1, pp. 99-122, 1997.
5. A. Shashua and T. R. Raviv, “The quotient image: Class based re-rendering and recognition with varying illuminations,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 129-139, 2001.
6. A. L. Yuille, D. Snow, Epstein R., and P. N. Belhumeur, “Determining generative models of objects under varying illumination: Shape and albedo from multiple images using SVD and integrability,” *International Journal of Computer Vision*, vol. 35, no. 3, pp. 203-222, 1999.
7. W. Zhao and R. Chellappa, “Symmetric shape from shading using self-ratio image,” *International Journal of Computer Vision*, vol. 45, pp. 55-752, 2001.
8. A. Georgiades, P. Belhumeur, and D. Kriegman, “From few to many: illumination cone models for face recognition under variable lighting and pose,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, pp. 643 -660, 2001.
9. R. T. Frankot and R. Chellappa, “A method for enforcing integrability in shape from shading problem,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-10, pp. 439-451, 1987.

10. D. Forsyth, "Shape from texture and integrability," *International Conference on Computer Vision*, pp. 447-453, 2001.
11. I. Shimshoni, Y. Moses, and M. Lindenbaum., "Shape reconstruction of 3D bilaterally symmetric surfaces," *International Journal of Computer Vision*, vol. 39, pp. 97-100, 2000.
12. T. Sim, S. Baker, and M. Bsat, "The CMU pose, illuminatin, and expression (PIE) database," *Prof. Face and Gesture Recognition*, pp. 53-58, 2002.
13. P. J. Philipps, H. Moon, P. Rauss, and S. Rivzi, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1090-1104, 2000.
14. W. T. Freeman and J. B. Tenenbaum, "Learning bilinear models for two-factor problems in vision," *IEEE Conf. on Computer Vision and Pattern Recognition*, 1997.
15. M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 72-86, 1991.
16. M.E. Brand, "Incremental singular value decomposition of uncertain data with missing values," *European Conference on Computer Vision*, pp. 707-720, 2002.
17. P. Belhumeur, D. Kriegman, and A. Yuille, "The bas-relief ambiguity," *International Journal of Computer Vision*, vol. 35, pp. 33-44, 1999.
18. V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1063-1074, 2003.
19. Q. F. Zheng and R. Chellappa, "Estimation of illuminant direction, albedo and shape from shading," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-13, pp. 680-702, 1991.
20. A. Shashua and A. Levin. "Linear Image Coding for Regression and Classification using the Tensor-rank Principle." *IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
21. C. Bregler, A. Hertzmann, and H. Biermann "Recovering Non-Rigid 3D Shape from Image Streams." *IEEE Conf. on Computer Vision and Pattern Recognition*, 2000.
22. M.A.O. Vasilescu and D. Terzopoulos, "Multilinear Analysis of Image Ensembles: TensorFaces," *European Conference on Computer Vision*, 2002.

Appendix-I: Recovering \mathbf{R} , \mathbf{F} , and \mathbf{S} from \mathbf{H}

This appendix presents an iterative algorithm that recovers \mathbf{R} , \mathbf{F} and \mathbf{S} from the observation matrix \mathbf{H} . In fact, we also infer $\mathbf{I} = [\Downarrow_{\mathbf{x}} [\Rightarrow_i \mathbf{i}_i(\mathbf{x})]]$, which is an indication matrix for \mathbf{H} .

We first concentrate on the most difficult part of updating \mathbf{R} with \mathbf{F} , \mathbf{S} , and \mathbf{I} fixed. We take vector derivatives of \mathcal{E} with respect to $\{r_{ij}; i = a, b, c; j = 1, \dots, m\}$ and treat the three terms in \mathcal{E} separately.

[About \mathcal{E}_0 .] With $\mathbf{f}_{j'} \doteq [\Downarrow_{j=1}^m f_{j'j}]$ and $\mathbf{s}_{j'} \doteq [s_{j'a}, s_{j'b}, s_{j'c}]^T$,

$$\frac{\partial \mathcal{E}_0}{\partial r_{ij}} = \sum_{j'=1}^n \sum_{\mathbf{x}=1}^d \mathbf{i}_{j'(\mathbf{x})} \{ \hat{\mathbf{w}}(\mathbf{x})^T \mathbf{R}(\mathbf{f}_{j'} \otimes \mathbf{s}_{j'}) - h_{j'(\mathbf{x})} \} \hat{\mathbf{w}}(\mathbf{x}) f_{j'j} s_{j'i}$$

$$\begin{aligned}
&= \sum_{j'=1}^n \sum_{\mathbf{x}=1}^d \mathbf{i}_{j'(\mathbf{x})} \left\{ \sum_{l=a,b,c} \sum_{k=1}^m \hat{\mathbf{w}}(\mathbf{x})^T \mathbf{r}_{lk} f_{j'k} s_{j'l} - h_{j'(\mathbf{x})} \right\} \hat{\mathbf{w}}(\mathbf{x}) f_{j'j} s_{j'i} \\
&= \sum_{l=a,b,c} \sum_{k=1}^m \left\{ \sum_{j'=1}^n \sum_{\mathbf{x}=1}^d \mathbf{i}_{j'(\mathbf{x})} f_{j'k} s_{j'l} f_{j'j} s_{j'i} \hat{\mathbf{w}}(\mathbf{x}) \hat{\mathbf{w}}(\mathbf{x})^T \right\} \mathbf{r}_{lk} \\
&\quad - \sum_{j'=1}^n \sum_{\mathbf{x}=1}^d \mathbf{i}_{j'(\mathbf{x})} h_{j'(\mathbf{x})} f_{j'j} s_{j'i} \hat{\mathbf{w}}(\mathbf{x}) \\
&= \sum_{l=a,b,c} \sum_{k=1}^m \mathbf{O}_{ij}^{lk} \mathbf{r}_{lk} - \gamma_{ij}, \tag{15}
\end{aligned}$$

where $\{\mathbf{O}_{ij}^{lk}; l = a, b, c; k = 1, \dots, m\}$ are properly defined $3m \times 3m$ matrices, and γ_{ij} is a properly defined $3m \times 1$ vector.

[About \mathcal{E}_1 .] Using forward differences to approximate partial derivatives ³,

$$\begin{aligned}
\frac{\partial \hat{a}_{j(x,y)}}{\partial y} &\simeq \hat{a}_{j(x,y+1)} - \hat{a}_{j(x,y)}; & \frac{\partial \hat{b}_{j(x,y)}}{\partial x} &\simeq \hat{b}_{j(x+1,y)} - \hat{b}_{j(x,y)}; \\
\frac{\partial \hat{c}_{j(x,y)}}{\partial x} &\simeq \hat{c}_{j(x+1,y)} - \hat{c}_{j(x,y)}; & \frac{\partial \hat{c}_{j(x,y)}}{\partial y} &\simeq \hat{c}_{j(x,y+1)} - \hat{c}_{j(x,y)},
\end{aligned} \tag{16}$$

we have

$$\alpha_{j(x,y)} \approx \hat{b}_{j(x+1,y)} \hat{c}_{j(x,y)} - \hat{b}_{j(x,y)} \hat{c}_{j(x+1,y)} + \hat{a}_{j(x,y)} \hat{c}_{j(x,y+1)} - \hat{a}_{j(x,y+1)} \hat{c}_{j(x,y)}. \tag{17}$$

Suppose we are given the product of albedo and surface normal as in (11), we can derive the albedo $p_j(\mathbf{x})$ and surface normals $\hat{a}_j(\mathbf{x})$, $\hat{b}_j(\mathbf{x})$, and $\hat{c}_j(\mathbf{x})$ as follows:

$$p_j(\mathbf{x}) = \sqrt{(\hat{\mathbf{w}}(\mathbf{x})^T \mathbf{r}_{aj})^2 + (\hat{\mathbf{w}}(\mathbf{x})^T \mathbf{r}_{bj})^2 + (\hat{\mathbf{w}}(\mathbf{x})^T \mathbf{r}_{cj})^2}, \tag{18}$$

$$\hat{a}_j(\mathbf{x}) = \frac{\hat{\mathbf{w}}(\mathbf{x})^T \mathbf{r}_{aj}}{p_j(\mathbf{x})}, \quad \hat{b}_j(\mathbf{x}) = \frac{\hat{\mathbf{w}}(\mathbf{x})^T \mathbf{r}_{bj}}{p_j(\mathbf{x})}, \quad \hat{c}_j(\mathbf{x}) = \frac{\hat{\mathbf{w}}(\mathbf{x})^T \mathbf{r}_{cj}}{p_j(\mathbf{x})}. \tag{19}$$

So, their partial derivatives with respect to \mathbf{r}_{aj} are

$$\frac{\partial \hat{a}_j(\mathbf{x})}{\partial \mathbf{r}_{aj}} = \frac{\hat{\mathbf{w}}(\mathbf{x})}{p_j(\mathbf{x})} - \hat{\mathbf{w}}(\mathbf{x})^T \mathbf{r}_{aj} \frac{\hat{\mathbf{w}}(\mathbf{x}) \hat{\mathbf{w}}(\mathbf{x})^T \mathbf{r}_{aj}}{p_j^3(\mathbf{x})} = \frac{1 - \hat{a}_j^2(\mathbf{x})}{p_j(\mathbf{x})} \hat{\mathbf{w}}(\mathbf{x}), \tag{20}$$

$$\frac{\partial \hat{a}_j(\mathbf{x})}{\partial \mathbf{r}_{bj}} = -\hat{\mathbf{w}}(\mathbf{x})^T \mathbf{r}_{aj} \frac{\hat{\mathbf{w}}(\mathbf{x}) \hat{\mathbf{w}}(\mathbf{x})^T \mathbf{r}_{bj}}{p_j^3(\mathbf{x})} = \frac{-\hat{a}_j(\mathbf{x}) \hat{b}_j(\mathbf{x})}{p_j(\mathbf{x})} \hat{\mathbf{w}}(\mathbf{x}), \quad \frac{\partial \hat{a}_j(\mathbf{x})}{\partial \mathbf{r}_{cj}} = \frac{-\hat{a}_j(\mathbf{x}) \hat{c}_j(\mathbf{x})}{p_j(\mathbf{x})} \hat{\mathbf{w}}(\mathbf{x}). \tag{21}$$

Similarly, we can derive their partial derivatives with respect to \mathbf{r}_{bj} and \mathbf{r}_{cj} , which are summarized as follows:

$$\frac{\partial \hat{k}_j(\mathbf{x})}{\partial \mathbf{r}_{lj}} = \frac{-\hat{k}_j(\mathbf{x}) \hat{l}_j(\mathbf{x})}{p_j(\mathbf{x})} \hat{\mathbf{w}}(\mathbf{x}), \quad \frac{\partial \hat{k}_j(\mathbf{x})}{\partial \mathbf{r}_{kj}} = \frac{1 - \hat{k}_j^2(\mathbf{x})}{p_j(\mathbf{x})} \hat{\mathbf{w}}(\mathbf{x}), \quad k, l \in \{a, b, c\}, \quad k \neq l. \tag{22}$$

³ Partial derivatives of boundary pixels require different approximations. But, similar derivations (skipped here due to space limitation) can be derived.

Notice that $\frac{\partial \hat{a}_{ij}(\mathbf{x})}{\partial \mathbf{r}_{bj}} = \frac{\partial \hat{b}_{ij}(\mathbf{x})}{\partial \mathbf{r}_{aj}}, \frac{\partial \hat{a}_{ij}(\mathbf{x})}{\partial \mathbf{r}_{cj}} = \frac{\partial \hat{c}_{ij}(\mathbf{x})}{\partial \mathbf{r}_{aj}},$ and $\frac{\partial \hat{b}_{ij}(\mathbf{x})}{\partial \mathbf{r}_{cj}} = \frac{\partial \hat{c}_{ij}(\mathbf{x})}{\partial \mathbf{r}_{bj}},$ which implies saving in computations.

We now compute the partial derivative of $\alpha_{j(x,y)}$ with respect to \mathbf{r}_{aj} :

$$\begin{aligned}
 \frac{\partial \alpha_{j(x,y)}}{\partial \mathbf{r}_{aj}} &= \frac{\partial}{\partial \mathbf{r}_{aj}} \{ \hat{b}_{j(x+1,y)} \hat{c}_{j(x,y)} - \hat{b}_{j(x,y)} \hat{c}_{j(x+1,y)} + \hat{a}_{j(x,y)} \hat{c}_{j(x,y+1)} \\
 &\quad - \hat{a}_{j(x,y+1)} \hat{c}_{j(x,y)} \} \\
 &= \left\{ \frac{\hat{a}_{j(x,y)} \hat{c}_{j(x,y)}}{p_{j(x,y)} p_{j(x,y+1)}} \hat{\mathbf{w}}_{(x,y)} \hat{\mathbf{w}}_{(x,y+1)}^T - \frac{\hat{a}_{j(x,y+1)} \hat{c}_{j(x,y+1)}}{p_{j(x,y)} p_{j(x,y+1)}} \hat{\mathbf{w}}_{(x,y+1)} \hat{\mathbf{w}}_{(x,y)}^T \right\} \mathbf{r}_{aj} \\
 &\quad + \left\{ \frac{\hat{a}_{j(x+1,y)} \hat{c}_{j(x+1,y)}}{p_{j(x,y)} p_{j(x+1,y)}} \hat{\mathbf{w}}_{(x+1,y)} \hat{\mathbf{w}}_{(x,y)}^T - \frac{\hat{a}_{j(x,y)} \hat{c}_{j(x,y)}}{p_{j(x,y)} p_{j(x+1,y)}} \hat{\mathbf{w}}_{(x,y)} \hat{\mathbf{w}}_{(x+1,y)}^T \right\} \mathbf{r}_{bj} \\
 &\quad + \left\{ \frac{\hat{a}_{j(x,y)} \hat{b}_{j(x,y)}}{p_{j(x,y)} p_{j(x+1,y)}} \hat{\mathbf{w}}_{(x,y)} \hat{\mathbf{w}}_{(x+1,y)}^T - \frac{\hat{a}_{j(x+1,y)} \hat{b}_{j(x+1,y)}}{p_{j(x,y)} p_{j(x+1,y)}} \hat{\mathbf{w}}_{(x+1,y)} \hat{\mathbf{w}}_{(x,y)}^T \right. \\
 &\quad \left. + \frac{1 - \hat{a}_{j(x,y)}^2}{p_{j(x,y)} p_{j(x,y+1)}} \hat{\mathbf{w}}_{(x,y)} \hat{\mathbf{w}}_{(x,y+1)}^T - \frac{1 - \hat{a}_{j(x,y+1)}^2}{p_{j(x,y)} p_{j(x+1,y)}} \hat{\mathbf{w}}_{(x,y+1)} \hat{\mathbf{w}}_{(x,y)}^T \right\} \mathbf{r}_{cj} \\
 &= \mathbf{P}_{aj(x,y)}^a \mathbf{r}_{aj} + \mathbf{P}_{aj(x,y)}^b \mathbf{r}_{bj} + \mathbf{P}_{aj(x,y)}^c \mathbf{r}_{cj} = \sum_{l=a,b,c} \mathbf{P}_{aj(x,y)}^l \mathbf{r}_{lj}, \quad (23)
 \end{aligned}$$

where $\mathbf{P}_{aj(x,y)}^a$, $\mathbf{P}_{aj(x,y)}^b$, and $\mathbf{P}_{aj(x,y)}^c$ are properly defined matrices of dimension $3m \times 3m$. By the same token, using properly defined $\mathbf{P}_{bj(x,y)}^a$, $\mathbf{P}_{bj(x,y)}^b$, $\mathbf{P}_{bj(x,y)}^c$, $\mathbf{P}_{cj(x,y)}^a$, $\mathbf{P}_{cj(x,y)}^b$, and $\mathbf{P}_{cj(x,y)}^c$, we can calculate $\frac{\partial \alpha_{j(x,y)}}{\partial \mathbf{r}_{ij}} = \sum_{l=a,b,c} \mathbf{P}_{ij(x,y)}^l \mathbf{r}_{lj}$ for $i = a, b, c$, and, finally,

$$\frac{\partial \mathcal{E}_1}{\partial \mathbf{r}_{ij}} = \sum_{\mathbf{x}=1}^d \alpha_{j(\mathbf{x})} \sum_{l=a,b,c} \mathbf{P}_{ij(\mathbf{x})}^l \mathbf{r}_{lj} = \sum_{l=a,b,c} \mathbf{P}_{ij}^l \mathbf{r}_{lj}; \quad \mathbf{P}_{ij}^l \doteq \sum_{\mathbf{x}=1}^d \alpha_{j(\mathbf{x})} \mathbf{P}_{ij(\mathbf{x})}^l. \quad (24)$$

[About \mathcal{E}_2 .] The symmetry constraint term $\beta_{j(\mathbf{x})}$ defined as in (13) can be expressed as

$$\beta_{j(\mathbf{x})}^2 = \mathbf{r}_{aj}^T \mathbf{Q}_{(\mathbf{x})}^a \mathbf{r}_{aj} + \mathbf{r}_{bj}^T \mathbf{Q}_{(\mathbf{x})}^b \mathbf{r}_{bj} + \mathbf{r}_{cj}^T \mathbf{Q}_{(\mathbf{x})}^c \mathbf{r}_{cj}, \quad (25)$$

where $\mathbf{Q}_{(\mathbf{x})}^a$, $\mathbf{Q}_{(\mathbf{x})}^b$, and $\mathbf{Q}_{(\mathbf{x})}^c$ are symmetric matrices with size $3m \times 3m$:

$$\mathbf{Q}_{(\mathbf{x})}^a = (\hat{\mathbf{w}}_{(\mathbf{x})} + \hat{\mathbf{w}}_{(\bar{\mathbf{x}})})(\hat{\mathbf{w}}_{(\mathbf{x})} + \hat{\mathbf{w}}_{(\bar{\mathbf{x}})})^T, \mathbf{Q}_{(\mathbf{x})}^b = (\hat{\mathbf{w}}_{(\mathbf{x})} - \hat{\mathbf{w}}_{(\bar{\mathbf{x}})})(\hat{\mathbf{w}}_{(\mathbf{x})} - \hat{\mathbf{w}}_{(\bar{\mathbf{x}})})^T, \mathbf{Q}_{(\mathbf{x})}^c = \mathbf{Q}_{(\mathbf{x})}^b. \quad (26)$$

The derivatives of $\beta_{j(\mathbf{x})}^2/2$ and \mathcal{E}_2 with respect to \mathbf{r}_{aj} , \mathbf{r}_{bj} , and \mathbf{r}_{cj} are

$$\frac{\partial \{\beta_{j(\mathbf{x})}^2/2\}}{\partial \mathbf{r}_{ij}} = \mathbf{Q}_{(\mathbf{x})}^i \mathbf{r}_{ij}; \quad \frac{\partial \mathcal{E}_2}{\partial \mathbf{r}_{ij}} = \sum_{\mathbf{x}=1}^d \mathbf{Q}_{(\mathbf{x})}^i \mathbf{r}_{ij} = \mathbf{Q}^i \mathbf{r}_{ij}; \quad \mathbf{Q}^i = \sum_{\mathbf{x}=1}^d \mathbf{Q}_{(\mathbf{x})}^i. \quad (27)$$

Putting the above derivations together and using $\frac{\partial \mathcal{E}}{\partial \mathbf{r}_{ij}} = 0$, we have

$$\sum_{l=a,b,c} \sum_{k=1}^m \mathbf{O}_{ij}^{lk} \mathbf{r}_{lk} + \lambda_1 \sum_{l=a,b,c} \mathbf{P}_{ij}^l \mathbf{r}_{lj} + \lambda_2 \mathbf{Q}^i \mathbf{r}_{ij} = \gamma_{ij}; \quad i = a, b, c; \quad j = 1, \dots, m. \quad (28)$$

We therefore arrive at a set of equations linear in $\{r_{ij}; i = a, b, c; j = 1, \dots, m\}$ that can be solved easily. After finding the new R , we normalize it using $R = R / \|R\|_2$.

We now illustrate how to update $F = [\Rightarrow_i f_i]$, $S = [\Rightarrow_i s_i]$, and $I = [\Rightarrow_i I_i]$ with R fixed (or W fixed). First notice that they are only involved in \mathcal{E}_0 . Moreover, f_i , s_i and I_i are related with only the image h_i . This becomes the same as the illumination separation problem defined in Section 4 and Appendix-II presents such a recovery algorithm, which also is iterative in nature. After running one iterative step to obtain the updated F , S , and I , we proceed to update R again and this process carries on until convergence.

Appendix-II: Recovering f and s from h given W

This recovery task is equivalent to minimizing the cost function defined as

$$\mathcal{E}_h(f, s) \doteq \|I \circ (h - W(f \otimes s))\|^2 + (1^T f - 1)^2, \quad (29)$$

where $I_{d \times 1}$ indicates the inclusion or exclusion of the pixels of the image h and \circ denotes the Hadamard (element-wise) product. Notice that (29) actually can be easily generalized as a cost function for robust estimation if the L_2 norm $\|\cdot\|$ is replaced by a robust function, and I by an appropriate weight function.

The following algorithm is an extension of bilinear analysis, with occlusion embedded. Firstly, we solve the least square (LS) solution f , given s and I .

$$f = \begin{bmatrix} W_f \\ 1^T \end{bmatrix}^\dagger \begin{bmatrix} I \circ h \\ 1 \end{bmatrix}; \quad W_f \doteq [\Rightarrow_i (T_i s)]_{d \times m}. \quad (30)$$

where $[\cdot]^\dagger$ denotes the pseudo-inverse. Secondly, we solve the LS solution s , given f and I :

$$s = W_s^\dagger (I \circ h); \quad W_s \doteq [\Rightarrow_i a_i]f, [\Rightarrow_i b_i]f, [\Rightarrow_i c_i]f]_{d \times 3} \doteq [Af, Bf, Cf], \quad (31)$$

where $A_{d \times m} \doteq [\Rightarrow_i a_i]$, $B_{d \times m} \doteq [\Rightarrow_i b_i]$, and $C_{d \times m} \doteq [\Rightarrow_i c_i]$ contain the information on the product of albedos and x , y , and z directions of the surface normals, respectively. In the third step, given f and s we update I as follows⁴:

$$I = [\|h - W(f \otimes s)\| < \eta], \quad (32)$$

where η is a pre-defined threshold.

Note that in Eqs. (30) and (31), additional saving in computation is possible. We can form matrices W'_f and W'_s and vector h' , with a reduced dimension, from W_f , W_s , and h , respectively, by discarding those rows corresponding to the excluded pixels and applying the primed version in (30) and (31).

For fast convergence, we use the following initial values in our implementation. We estimate s using the algorithm presented in [19] and set I to exclude those pixels whose intensities are smaller than a certain threshold.

⁴ This is a Matlab operation which performs an element-wise comparison.

User Assisted Separation of Reflections from a Single Image Using a Sparsity Prior

Anat Levin and Yair Weiss

School of Computer Science and Engineering
The Hebrew University of Jerusalem
91904 Jerusalem, Israel
{alevin,yweiss}@cs.huji.ac.il

Abstract. When we take a picture through transparent glass the image we obtain is often a linear superposition of two images: the image of the scene beyond the glass plus the image of the scene reflected by the glass. Decomposing the single input image into two images is a massively ill-posed problem: in the absence of additional knowledge about the scene being viewed there are an infinite number of valid decompositions. In this paper we focus on an easier problem: user assisted separation in which the user interactively labels a small number of gradients as belonging to one of the layers.

Even given labels on part of the gradients, the problem is still ill-posed and additional prior knowledge is needed. Following recent results on the statistics of natural images we use a sparsity prior over derivative filters. We first approximate this sparse prior with a Laplacian prior and obtain a simple, convex optimization problem. We then use the solution with the Laplacian prior as an initialization for a simple, iterative optimization for the sparsity prior. Our results show that using a prior derived from the statistics of natural images gives a far superior performance compared to a Gaussian prior and it enables good separations from a small number of labeled gradients.

1 Introduction

Figure 1(a) shows the room in which Leonardo’s Mona Lisa is displayed at the Louvre. In order to protect the painting, the museum displays it behind a transparent glass. While this enables viewing of the painting, it poses a problem for the many tourists who want to photograph the painting (see figure 1(b)). Figure 1(c) shows a typical picture taken by a tourist¹: the wall across from the painting is reflected by the glass and the picture captures this reflection superimposed on the Mona-Lisa image.

A similar problem occurs in various similar settings: photographing window dressings, jewels and archaeological items protected by glass. Professional photographers attempt to solve this problem by using a polarizing lens. By rotating

¹ All three images are taken from www.studiolo.org/Mona/MONA09.htm

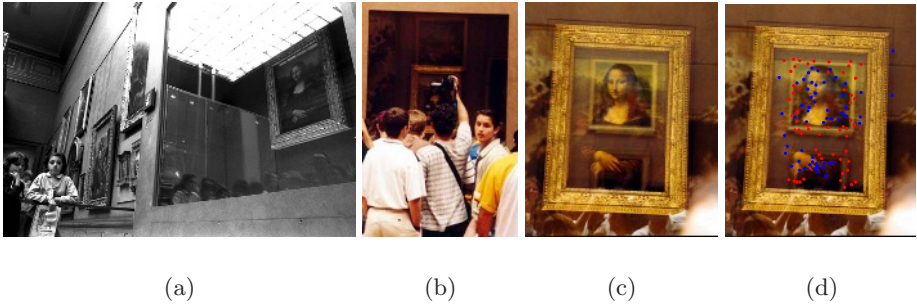


Fig. 1. (a),(b) The scene near the Mona Lisa in the Louvre. The painting is housed behind glass to protect it from the many tourists. (c) A photograph taken by a tourist at the Louvre. The photograph captures the painting as well as the reflection of the wall across the room. (d) The user assisted reflection problem. We assume the user has manually marked gradients as belonging to the painting layer or the reflection layer and wish to recover the two layers.

the polarizing lens appropriately, one can reduce (but not eliminate) the reflection. As suggested in [2,8] the separation can be improved by capturing two images with two different rotations of the polarizing lens and taking an optimal linear combination of the two images. An alternative solution is to use *multiple* input images [11,4] in which the reflection and the non-reflected images have different motions. By analyzing the movie sequence, the two layers can be recovered. In [13], a similar approach is applied to stereo pairs.

While the approaches based on polarizing lenses or stereo images may be useful for professional photographers, they seem less appealing for a consumer-level application. Viewing the image in figure 1(c), it seems that the information for the separation is present in a single image. Can we use computer vision to separate the reflections from a single image ?

Mathematically, the problem is massively ill-posed. The input image $I(x, y)$ is a linear combination of two unknown images $I_1(x, y), I_2(x, y)$:

$$I(x, y) = I_1(x, y) + I_2(x, y) \quad (1)$$

Obviously, there are an infinite number of solutions to equation 1: the number of unknowns is twice the number of equations. Additional assumptions are needed. On the related problem of separating shading and reflectance, impressive results have been obtained using a single image [12,3]. These approaches make use of the fact that edges due to shading and edges due to reflectance have different statistics (e.g. shading edges tend to be monochromatic). Unfortunately, in the case of reflections, the two layers have the same statistics, so the approaches used for shading and reflectance are not directly applicable. In [5], a method was presented that used a prior on images to separate reflections with no user intervention. While impressive results were shown on simple images, the technique used a complicated optimization that often failed to converge on complex images.

In this paper, we present a technique that works on arbitrarily complex images but we simplify the problem by allowing user assistance. We allow the user to *manually* mark certain edges (or areas) in the image as belonging to one of the two layers. Figure 1(d) shows the Mona Lisa image with manually marked gradients: blue gradients are marked as belonging to the Mona Lisa layer and red are marked as belonging to the reflection layer. The user can either label individual gradients or draw a polygon to indicate that all gradients inside the polygon belong to one of the layers. This kind of user assistance seems quite natural in the application we are considering: imagine a Photoshop plugin that a tourist can use to post-process the images taken with reflections. As long as the user needs only to mark a small number of edges, this seems a small price to pay.

Even when the user marks a small number of edges, the problem is still ill-posed. Consider an image with a million pixels and assume the user marks a hundred edges. Each marked edge gives an additional constraint for the problem in equation 1. However, with these additional equations, the total number of equations is only a million and a hundred, far less than the two million unknowns. Unless the user marks every single edge in the image, additional prior knowledge is needed.

Following recent studies on the statistics of natural scenes [7,9], we use a prior on images that is based on the sparsity of derivative filters. We first approximate this prior with a Laplacian prior and this approximation enables us to find the most likely decomposition using *convex* optimization. We then use the Laplacian prior solution as an initial guess for a simple, iterative optimization of the sparsity prior. We show that by using a prior derived from the statistics of natural scenes, one can obtain excellent separations using a small number of labeled gradients.

2 Statistics of Natural Images

A remarkably robust property of natural images that has received much attention lately is the fact that when derivative filters are applied to natural images, the filter outputs tend to be sparse [7,9,17]. Figure 2(a-d) illustrates this fact: the histogram of the vertical derivative filter is peaked at zero and fall off much faster than a Gaussian. These distributions are often called “sparse” and there are a number of ways to formulate this property mathematically, (e.g. in terms of their tails or their kurtosis).

We will follow Mallat [6] and Simoncelli [10] in characterizing these distributions in terms of the shape of their logarithm. As shown in figure 2(b,d), when we look at the logarithm of the histogram the curve is always below the straight line connecting the maximum and minimum values. This should be contrasted with the Gaussian distribution (that is always above the straight line) or the Laplacian distribution (that is simply a straight line in the log domain) (figure 2(e)). In [5] it was shown that the fact that the log distribution is always below the straight line, is crucial for obtaining transparency decompositions from a single

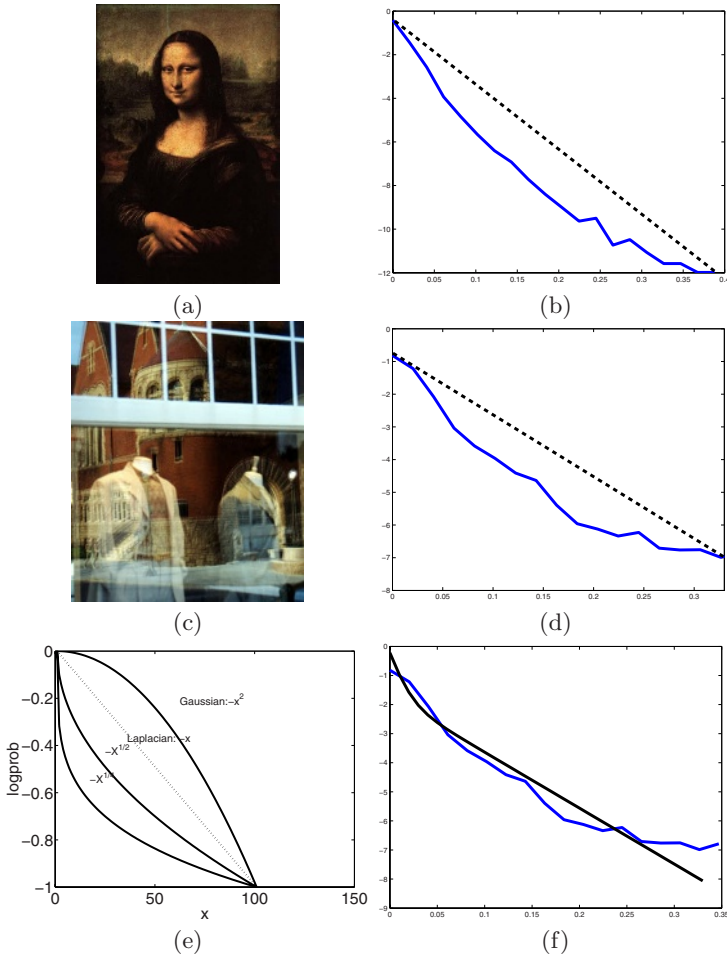


Fig. 2. (a),(c) input images. (b),(d) log-histogram of d_y derivative. A robust property of natural images is that the log-histograms of derivative filters lie below the straight line connecting the minimal and maximal values. We refer to such distributions as sparse (e) Log probabilities for distributions of the form e^{-x^α} . The Gaussian distribution is not sparse (it is always above the straight line) and distributions for which $\alpha < 1$ are sparse. The Laplacian distribution is exactly at the border between sparse and non sparse distributions. (f) Matching a mixture model to a filter output histogram. The mixture parameters were selected to maximize the likelihood of the histogram. A mixture of Laplacians is sparse even though the individual components are not.

image. Distributions that are above the straight line will prefer to split an edge of unit contrast into two edges (one in each layer) with half the contrast, while distributions below the line will prefer decompositions in which the edge only appears in one of the layers but not in the other. We will refer to distributions that have this property in the log domain as being sparse.

Wainwright and Simoncelli [15] have suggested describing the histograms of natural images with an infinite Gaussian mixture model. By adding many Gaussians, each with a mean at zero but with different variances one can obtain sparse distributions. This can also be achieved by mixing only two distributions: a narrow distribution centered on zero and a broad distribution centered on zero will give a sparse distribution. Figure 2(f) shows a mixture of two Laplacian distributions:

$$\Pr(x) = \frac{\pi_1}{2s_1} e^{-|x|/s_1} + \frac{\pi_2}{2s_2} e^{-|x|/s_2} \quad (2)$$

Although the Laplacian distributions are not sparse based on our definition, the mixture is. For the experiments in this paper, the mixture parameters were learned from real images. That is, the parameters were selected to maximize the likelihood of the histogram of derivative filters, as in Figure 2(f). The learned values we found are $s_1 = 0.01$, $s_2 = 0.05$, $\pi_1 = 0.4$, $\pi_2 = 0.6$.

Given the histograms over derivative filters, we follow [16] in using it to define a distribution over images by assuming that derivative filters are independent over space and orientation so that our prior over images is given by:

$$\Pr(I) = \prod_{i,k} \Pr(f_{i,k} \cdot I) \quad (3)$$

where $f \cdot I$ denotes the inner product between a linear filter f and an image I , and $f_{i,k}$ is the k 'th derivative filter centered on pixel i . The derivative filters set we use includes two orientations (horizontal and vertical) and two degrees (i.e. first derivative filters as well as second derivative). The probability of a single derivative is given by equation 2.

Equation 3 gives the probability of a single layer. We follow [5] in defining the probability of a decomposition I^1, I^2 as the product of the probabilities of each layer (i.e. assuming the two layers are independent).

3 Optimization

We are now ready to state the problem formally. We are given an input image I and two sets of image locations S_1, S_2 so that gradients in location S_1 belong to layer 1 and gradients in location S_2 belong to layer 2. We wish to find two layers I_1, I_2 such that:

1. the two layers sum to form the input image $I = I_1 + I_2$
2. the gradients of I_1 at all locations in S_1 agree with the gradients of the input image I and similarly the gradients of I_2 at all locations in S_2 agree with the gradients of I .

Subject to these two constraints we wish to maximize the probability of the layers $\Pr(I^1, I^2) = \Pr(I^1) \Pr(I^2)$ given by equation 3.

Our approximation proceeds in two steps. We first approximate the sparse distribution with a Laplacian prior. This leads to a *convex* optimization problem for which the global maximum can be found using linear programming. We then use the solution with a Laplacian prior as an initial condition for a simple, iterative maximization of the sparse prior.

3.1 Exactly Maximizing a Laplacian Prior Using Linear Programming

Under the Laplacian approximation, we approximate $\Pr(I)$ with an approximate $\tilde{\Pr}(I)$ defined as:

$$Pr(I) = \prod_{i,k} e^{-|f_{i,k} \cdot I|} \quad (4)$$

To find the best decomposition under the Laplacian approximation we need to minimize:

$$J(I_1, I_2) = \sum_{i,k} |f_{i,k} \cdot I_1| + |f_{i,k} \cdot I_2| \quad (5)$$

subject to the two constraints given above: that $I_1 + I_2 = I$ and that the two layers agree with the labeled gradients. This is an L_1 minimization with linear constraints. We can turn this into an unconstrained minimization by substituting in $I_2 = I - I_1$ so that we wish to find a single layer I^1 that minimizes:

$$\begin{aligned} J_2(I_1) = & \sum_{i,k} |f_{i,k} \cdot I_1| + |f_{i,k} \cdot (I - I_1)| \\ & + \lambda \sum_{i \in S_1, k} |f_{i,k} \cdot I_1 - f_{i,k} \cdot I| \\ & + \lambda \sum_{i \in S_2, k} |f_{i,k} \cdot I_1| \end{aligned} \quad (6)$$

where the last two terms enforce the agreement with the labeled gradients.

This minimization can be performed exactly using linear programming. This is due to the fact that the derivatives are linear functions of the unknown image. To see this, define v to be a vectorized version of the image I_1 then we can rewrite J_2 as:

$$J_2(v) = \|Av - b\|_1 \quad (7)$$

where $\|\cdot\|_1$ is the L_1 norm, the matrix A has rows that correspond to the derivative filters and the vector b either has input image derivatives or zero so that equation 7 is equivalent to equation 6.

Minimization of equation 7 can be done by introducing slack variables and solving:

$$\begin{aligned} \text{Min : } & \sum_i (z_i^+ + z_i^-) \\ \text{Subject to : } & \\ & Av + (z^+ - z^-) = b \\ & z^+ \geq 0, z^- \geq 0 \end{aligned}$$

The idea is that at the optimal solution one of the variables z_i^+, z_i^- is zero, and the over is equal to $|A_{i \rightarrow v} - b_i|$. The above problem is a standard linear programming one and we use the LOQO [14] linear programming package to solve it.

3.2 Optimization of the Sparse Prior Using Iterated Linear Programming

To find the most likely decomposition under the sparse prior we need to maximize the probability of the two layers as given by equation 3. Using the same algebra as in the previous section this is equivalent to finding a vector v that minimizes:

$$J_3(v) = \sum_i \rho(A_{i \rightarrow} v - b_i) \quad (8)$$

where $\rho(x)$ is the log probability shown in figure 2. $\rho(x)$ is similar to a robust error measure and hence minimizing J_3 is *not* a convex optimization problem. Nevertheless, using EM we can iteratively solve convex problems.

Since we use a mixture model to describe the sparse prior, we can use expectation-maximization (EM) [1] to iteratively improve the probability of a decomposition. We introduce a binary hidden variable h_i for every row of the matrix A that denotes which Laplacian generated the measurement in b_i . In the E step we calculate the expectation of h_i and in the M step we use this expected value and optimize an expected complete data log likelihood. A standard derivation shows that the EM algorithm reduces to:

- E step. calculate two weights w_1, w_2 for every row of the matrix A :

$$w_j(i) \propto \frac{\pi_j}{s_j} e^{-|A_{i \rightarrow} v - b_i|/s_j} \quad (9)$$

the proportion constant is set so that $w_1(i) + w_2(i) = 1$ for all i .

- M step: perform an L_1 minimization given by:

$$v^* \leftarrow \arg \min_v \|DAv - Db\|_1 \quad (10)$$

with D a diagonal matrix whose elements are given by:

$$D(i, i) = w_1(i)/s_1 + w_2(i)/s_2 \quad (11)$$

At every iteration, we are provably decreasing the cost function J_3 in equation 8. The optimization in the M step was performed using the same linear programming software as in the Laplacian approximation. 3 EM iterations are usually sufficient.

4 Results

We show results of our algorithm on five images of scenes with reflections. Four of the images were downloaded from the internet and we had no control over the camera parameters or the compression methods used. For color images we ran the algorithm separately on the R, G and B channels.

Figures 3, 4 and 5 show the input images with labeled gradients, and our results. In Figures 4,5 we compare the Laplacian prior and the sparse prior,



Fig. 3. Results: (a) input image. (b-c) decomposition.

versus the number of labeled points. The Laplacian prior gives good results although some ghosting effects can still be seen (i.e. there are remainders of layer 2 in the reconstructed layer 1). These ghosting effects are fixed by the sparse prior. Good results can be obtained with a Laplacian prior when more labeled gradients are provided. Figures 6, 7 compares the Laplacian prior with a Gaussian prior (i.e. minimizing $\|Av - b\|$ under the L_2 norm) using both simple and real images. The non sparse nature of the Gaussian distribution is highly noticeable, causing the decomposition to split edges into two low contrast edges, rather than putting the entire contrast in one of the layers.



Fig. 4. Comparing Laplacian prior (first iteration results) with a sparse prior. When a few gradients are labeled (left) the sparse prior gives noticeably better results. When more gradients are labeled (right), the Laplacian prior results are similar to the sparse prior. (a-b) labeled input images. (c-d) decomposition with Laplacian prior. (e-f) decomposition using a sparse prior.

The images in figure 5 were separated automatically in [11] using multiple images. An advantage of using multiple images is that they can deal better with saturated regions (e.g. the cheekbone of the man in the image that is superimposed on the white shirt of the woman) since the saturated region location varies along the sequence. However, working with a single image, we cannot recover structure in saturated regions.

In Fig 8 the technique was applied for removing shading artifacts. For this problem, the same algorithm was applied in the log-domain.

5 Discussion

Separating reflections from a single image is a massively ill-posed problem. In this paper we have focused on slightly easier problem in which the user marks a small number of gradients as belonging to one of the layers. This is still an ill-posed problem and we have used a prior derived from the statistics of natural scenes: that derivative filters have sparse distributions. We showed how to efficiently find the most probable decompositions under this prior using linear programming. Our results show the clear advantage of a technique that is based on natural scene statistics rather than simply assuming a Gaussian distribution.

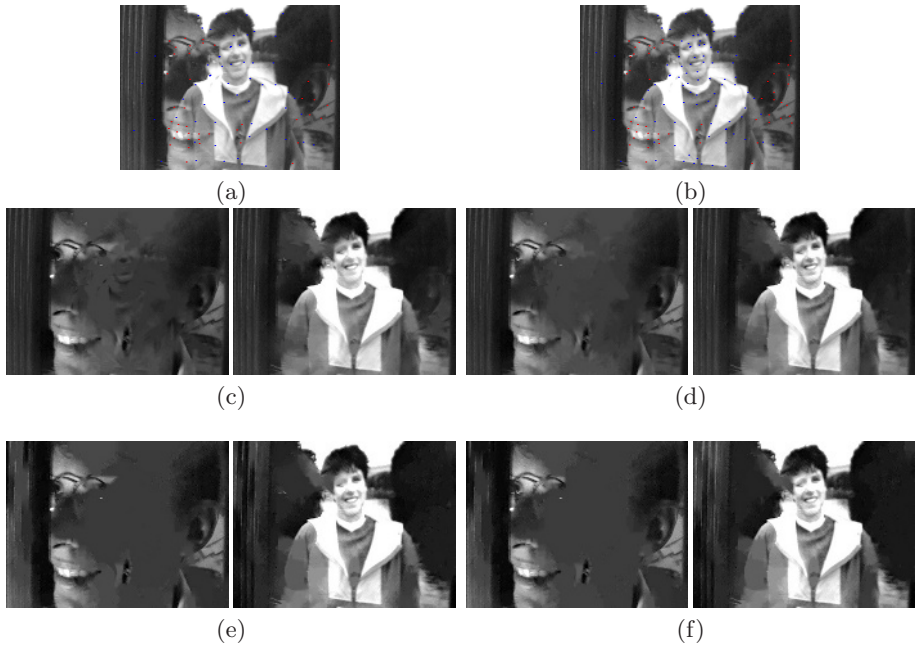


Fig. 5. Comparing Laplacian prior (first iteration results) with a sparse prior. When a few gradients are labeled (left) the sparse prior gives noticeably better results. When more gradients are labeled (right), the Laplacian prior results are similar to the sparse prior. (a-b) labeled input images. (c-d) decomposition with Laplacian prior. (e-f) decomposition using a sparse prior.

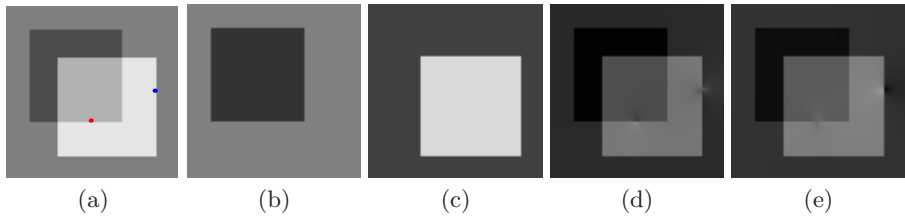


Fig. 6. (a) A very simple image with two labeled points. (b-c) The Laplacian prior gives the correct results for this image while the Gaussian prior (c-d) does not. The Gaussian prior prefers to split edges into two low contrast edges.

Since we are using an off-the-shelf linear programming package, we are not taking advantage of the spatial properties of the optimization problem. The current run time of the linear programming for images of size 240×320 is a few minutes on a standard PC. We have not performed an extensive comparison of linear programming packages so that with other packages the run times may be significantly faster. We are currently working on deriving specific algorithms for minimizing L_1 cost functions on image derivatives. Since this is a convex



Fig. 7. Gaussian prior results: (a) results on the second column of fig4. (b) results on the second column of fig5.

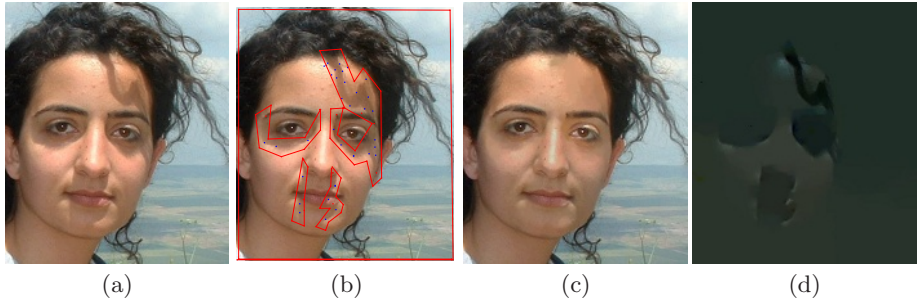


Fig. 8. Removing shading artifacts (a) original image. (b) labeled image. (c-d) decomposition

problem, local minima are not an issue and so a wide range of iterative algorithms may be used. In preliminary experiments, we have found that a multigrid algorithm can minimize such cost functions significantly faster. We are also investigating using a mixture of Gaussians rather than a mixture of Laplacians to describe sparse distributions. This leads to M steps in which L_2 minimizations need to be performed, and there are a wide range of efficient solvers for such minimizations.

We are also investigating the use of other features other than derivatives to describe the statistics of natural images. Our experience shows that when stronger statistical models are used, we need less labeled points to achieve a good separation. We hope that using more complex statistical models will still enable us to perform optimization efficiently. This may lead to algorithms that separate reflections from a single image, without any user intervention.

References

1. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:1–38, 1977.
2. H. Farid and E.H. Adelson. Separating reflections from images by use of independent components analysis. *Journal of the optical society of america*, 16(9):2136–2145, 1999.
3. G. D. Finlayson, S. D. Hordley, and M. S. Drew. removing shadows from images. In *European Conf. on Computer Vision*, 2002.

4. M. Irani and S. Peleg. Image sequence enhancement using multiple motions analysis. In *Conf. on Computer Vision and Pattern Recognition*, pages 216–221, Champaign, Illinois, June 1992.
5. A. Levin, A. Zomet, and Y. Weiss. Learning to perceive transparency from the statistics of natural scenes. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, 2002.
6. S. Mallat. A theory for multiresolution signal decomposition : the wavelet representation. *IEEE Trans. PAMI*, 11:674–693, 1989.
7. B.A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–608, 1996.
8. Y. Shechner, J. Shamir, and N. Kiryati. Polarization-based decorrelation of transparent layers: The inclination angle of an invisible surface. In *Int. Conf. on Computer Vision*, pages 814–819, 1999.
9. E.P. Simoncelli. Statistical models for images:compression restoration and synthesis. In *Proc Asilomar Conference on Signals, Systems and Computers*, pages 673–678, 1997.
10. E.P. Simoncelli. Bayesian denoising of visual images in the wavelet domain. In P Müller and B Vidakovic, editors, *Wavelet based models*, 1999.
11. R. Szeliksi, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *Conf. on Computer Vision and Pattern Recognition*, 2000.
12. M. Tappen, W.T. Freeman, and E.H. Adelson. Recovering intrinsic images from a single image. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, 2002.
13. Y. Tsin, S.B. Kang, and R. Szeliski. Stereo matching with reflections and translucency. In *Conf. on Computer Vision and Pattern Recognition*, pages 702–709, 2003.
14. R. Vanderbei. Loqo, <http://www.princeton.edu/~rvdb/>, 2000.
15. M.J. Wainwright, E.P. Simoncelli, and A.S. Willsky. Random cascades of gaussian scale mixtures for natural images. In *Int. Conf. on Image Processing*, pages I:260–263, 2000.
16. Y. Weiss. Deriving intrinsic images from image sequences. In *Proc. Intl. Conf. Computer Vision*, pages 68–75. 2001.
17. M. Zibulevsky, P. Kisilev, Y. Zeevi, and B. Pearlmutter. Blind source separation via multinode sparse representation. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, 2001.

The Quality of Catadioptric Imaging – Application to Omnidirectional Stereo

W. Stürzl, H.-J. Dahmen, and H.A. Mallot

Department of Cognitive Neuroscience, University of Tübingen
Auf der Morgenstelle 28, 72076 Tübingen, Germany
`wolfgang.stuerzl@uni-tuebingen.de`

Abstract. We investigate the influence of the mirror shape on the imaging quality of catadioptric sensors. For axially symmetrical mirrors we calculate the locations of the virtual image points considering incident quasi-parallel light rays. Using second order approximations, we give analytical expressions for the two limiting surfaces of this “virtual image zone”. This is different to numerical or ray tracing approaches for the estimation of the blur region, e.g. [1]. We show how these equations can be used to estimate the image blur caused by the shape of the mirror. As examples, we present two different omnidirectional stereo sensors with single camera and equi-angular mirrors that are used on mobile robots. To obtain a larger stereo baseline one of these sensors consists of two separated mirror of the same angular magnification and differs from a similar configuration proposed by Ollis et al. [2]. We calculate the caustic surfaces and show that this stereo configuration can be approximated by two single view points yielding an effective vertical stereo baseline of approx. 3.7 cm. An example of panoramic disparity computation using a physiologically motivated stereo algorithm is given.

1 Introduction

Omnidirectional catadioptric image sensors have become widely used in recent years [3]. Usually, the shape of a reflective surface is calculated considering only the reflection of the principal ray which runs through the nodal point of the camera lens (pinhole camera model). However, due to the camera lens and its aperture a finite area on the reflective surface contributes to the imaging of a single object point. Therefore the shape of the mirror can cause image blur, for example.

To investigate the imaging quality of a catadioptric system we first estimate the location of the virtual image of distant objects generated by a convex reflective surface (section 2). Using these results we then (section 3) calculate the expected image blur, which depends also on the focal length and the aperture of the lens. The intention of this paper is to formulate an approximative description of how the shape of the reflective surface may influence the imaging quality of a catadioptric system.

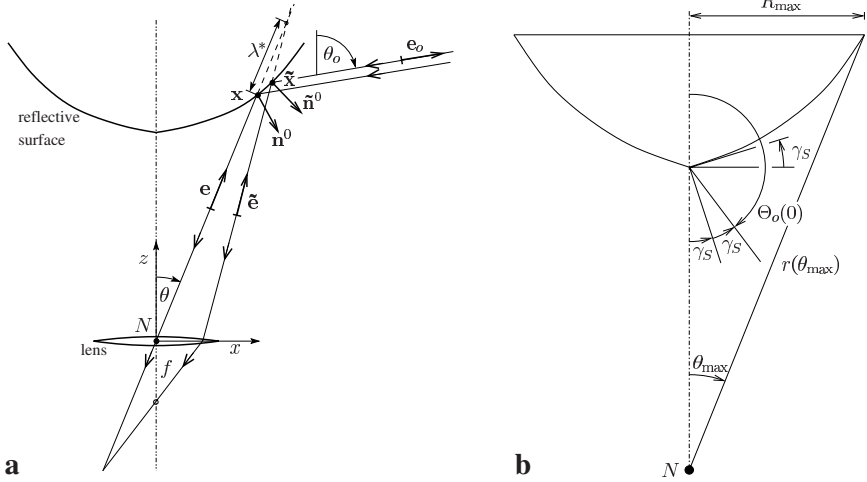


Fig. 1. a: Calculation of the virtual image zone: Two vertically shifted parallel rays with incidence direction $-\mathbf{e}_o$ originating from a distant point are reflected at neighboring points \mathbf{x} and $\tilde{\mathbf{x}}$. The corresponding virtual image point is the intersection of the reflected rays with direction vectors $-\mathbf{e}$ and $-\tilde{\mathbf{e}}$. The imaging is more complex for horizontally shifted parallel rays. **b:** Illustration of the parameters used in Eqs. (18) and (20), which determine – in addition to the magnification factor α – the shape of the reflective surface: γ_S is the angle relative to the x -axis at the tip, θ_{\max} is the camera angle corresponding to the maximum radius R_{\max} of the mirror. From the law of reflection, we find $\Theta_o(0) = \pi - 2\gamma_S$.

2 Calculation of the Virtual Image Zone

To compute the location of the virtual image of an object point we consider two incident parallel light rays¹ with direction $-\mathbf{e}_o$ which hit the mirror surface at the neighboring points \mathbf{x} and $\tilde{\mathbf{x}}$ and are reflected into directions $-\mathbf{e}$ and $-\tilde{\mathbf{e}}$, see Fig. 1 a. Therefore, the reflected rays can be described by $(\lambda, \mu \in \mathbb{R})$

$$\mathbf{g}(\lambda) = \mathbf{x} + \lambda \mathbf{e} \quad , \quad \tilde{\mathbf{g}}(\mu) = \tilde{\mathbf{x}} + \mu \tilde{\mathbf{e}} \quad . \quad (1)$$

The distance $\|\mathbf{g}(\lambda) - \tilde{\mathbf{g}}(\mu)\|^2$ is minimized for

$$\lambda^* = \frac{(\tilde{\mathbf{x}} - \mathbf{x})((\mathbf{e}\tilde{\mathbf{e}})\tilde{\mathbf{e}} - \mathbf{e})}{(\mathbf{e}\tilde{\mathbf{e}})^2 - 1} \quad , \quad \mu^* = \frac{(\mathbf{x} - \tilde{\mathbf{x}})((\mathbf{e}\tilde{\mathbf{e}})\mathbf{e} - \tilde{\mathbf{e}})}{(\mathbf{e}\tilde{\mathbf{e}})^2 - 1} \quad . \quad (2)$$

λ^* and μ^* determine the location of the “virtual image zone”. Whereas an intersection point exists for every two vertically shifted parallel rays, this is not true in general.

¹ We assume that the distance to the object is large compared to the radius of local curvature.

If \mathbf{g} in Eq. (1) describes the principal ray, and the mirror surface is given by $\mathbf{x}(\theta, \varphi)$ with respect to the nodal point of the camera then \mathbf{g} hits the mirror in the point $\mathbf{x} := \mathbf{x}(\theta, \varphi)$, where θ and φ are determined by the law of reflection, i.e.

$$\mathbf{e} = \mathbf{e}_o - 2(\mathbf{n}^0 \mathbf{e}_o) \mathbf{n}^0 . \quad (3)$$

$\mathbf{n}^0 := \mathbf{n}^0(\theta, \varphi)$ is the normal vector of the mirror surface at \mathbf{x} and

$$\mathbf{e} := \mathbf{e}(\theta, \varphi) := (\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta)^\top \quad (4)$$

is the unit direction vector expressed in spherical coordinates $\theta \in [0, \pi]$, $\varphi \in [0, 2\pi)$. For the second ray, which hits the mirror in the vicinity of \mathbf{x} , we get

$$\tilde{\mathbf{x}} = \mathbf{x}(\theta + \Delta\theta, \varphi + \Delta\varphi) , \quad (5)$$

$$\tilde{\mathbf{e}} = \mathbf{e}_o - 2(\tilde{\mathbf{n}}^0 \mathbf{e}_o) \tilde{\mathbf{n}}^0 \quad (6)$$

$$= \mathbf{e}_o - 2[\mathbf{n}^0(\theta + \Delta\theta, \varphi + \Delta\varphi) \mathbf{e}_o] \mathbf{n}^0(\theta + \Delta\theta, \varphi + \Delta\varphi) . \quad (7)$$

We found an approximate solution $\lambda^*(\theta)$ for axially symmetrical mirrors, for which the surface is described by

$$\mathbf{x}(\theta, \varphi) = r(\theta) \mathbf{e}(\theta, \varphi) . \quad (8)$$

$r(\theta)$ is the distance of a point on the mirror to the nodal point at the origin of the coordinate system. The normal vector at point $\mathbf{x}(\theta, \varphi)$ is then given by

$$\mathbf{n}^0(\theta, \varphi) = - (r(\theta)^2 + r'(\theta)^2)^{-\frac{1}{2}} (r(\theta) \mathbf{e}(\theta, \varphi) - r'(\theta) \frac{\partial}{\partial \theta} \mathbf{e}(\theta, \varphi)) . \quad (9)$$

By Taylor expansion of (5) and (7) with respect to $\Delta\theta$ and $\Delta\varphi$ up to second order an approximate estimate of (2) can be found analytically.² As shown in the appendix the result is

$$\lambda^* \approx \lambda_{\theta\varphi}^*(\theta, \Delta\theta, \Delta\varphi) \quad (10)$$

$$:= -\frac{1}{2} r(\theta) \left[1 + \left(\frac{r'(\theta)}{r(\theta)} \right)^2 \right] \frac{\Delta\theta^2 g(\theta) + \Delta\varphi^2 h(\theta) \sin \theta}{\Delta\theta^2 g(\theta)^2 + \Delta\varphi^2 h(\theta)^2} , \quad (11)$$

$$g(\theta) := 1 + 2 \left(\frac{r'(\theta)}{r(\theta)} \right)^2 - \frac{r''(\theta)}{r(\theta)} , \quad h(\theta) := \sin \theta - \frac{r'(\theta)}{r(\theta)} \cos \theta . \quad (12)$$

² Usually only a small part of the mirror contributes to the imaging of a single point: For a planar reflective surface, the angular extent $\Delta\theta$ of the mirror that contributes to the imaging at angle θ is approx. $\frac{D \cos \theta}{2r(\theta)}$, where D is the lens aperture. For a convex mirror of angular magnification α (see 2.1), $\Delta\theta$ is approximately reduced by the factor α^{-1} . We exclude rapid changes in curvature and nearly tangential incidence.

For vertically shifted incident parallel rays in an axial plane the reflected rays form a caustic curve. In this case the point given by $\lambda_{\theta\varphi}^*(\Delta\varphi = 0)$ is the tangent point of the principal ray $\mathbf{g}(\lambda)$ on the caustic. The existence of a caustic curve is not always guaranteed [4], since e.g. for horizontally shifted parallel incident rays $\det(\frac{\partial}{\partial \varphi} \tilde{\mathbf{x}}, \tilde{\mathbf{e}}, \frac{\partial}{\partial \varphi} \tilde{\mathbf{e}})$ may be $\neq 0$.

If the second ray is shifted solely vertically ($\Delta\varphi = 0$) or solely horizontally ($\Delta\theta = 0$) we obtain two special cases,

$$\lambda_{\theta}^*(\theta) := \lambda_{\theta\varphi}^*(\theta, \Delta\theta, \Delta\varphi = 0) = -\frac{1}{2}r(\theta) \frac{1 + \left(\frac{r'(\theta)}{r(\theta)}\right)^2}{1 + 2\left(\frac{r'(\theta)}{r(\theta)}\right)^2 - \frac{r''(\theta)}{r(\theta)}} , \quad (13)$$

$$\lambda_{\varphi}^*(\theta) := \lambda_{\theta\varphi}^*(\theta, \Delta\theta = 0, \Delta\varphi) = -\frac{1}{2}r(\theta) \frac{[1 + \left(\frac{r'(\theta)}{r(\theta)}\right)^2] \sin \theta}{\sin \theta - \frac{r'(\theta)}{r(\theta)} \cos \theta} , \quad (14)$$

which are the extremal values of (11) for fixed θ .³ The corresponding virtual image surfaces of the mirror are given by

$$\mathbf{v}_{\theta}(\theta, \varphi) := \mathbf{x}(\theta, \varphi) + \lambda_{\theta}^*(\theta) \mathbf{e}(\theta, \varphi) , \quad (15)$$

$$\mathbf{v}_{\varphi}(\theta, \varphi) := \mathbf{x}(\theta, \varphi) + \lambda_{\varphi}^*(\theta) \mathbf{e}(\theta, \varphi) . \quad (16)$$

If the difference between $\lambda_{\theta}^*(\theta)$ and $\lambda_{\varphi}^*(\theta)$ is large then the camera lens cannot be focused on both \mathbf{v}_{θ} and $\mathbf{v}_{\varphi}(\theta, \varphi)$, and the resulting image will be blurred.⁴ If the plane of focus is moved continuously, either horizontal or vertical structures will be imaged sharply at some points. Of course, the extent of blur also depends on the aperture of the camera lens. This will be discussed in section 3.

2.1 Mirrors with Equi-angular Magnification

Using the parameters γ_S , θ_{\max} and R_{\max} that are depicted in Fig. 1 b, the surface of a mirror with constant (vertical) angular magnification α , i.e.

$$\theta_o = \Theta_o(\theta) := \pi - 2\gamma_S - \alpha\theta \quad (17)$$

($\Rightarrow |\frac{\partial}{\partial\theta}\Theta_o(\theta)| = \alpha$), can be described by

$$r(\theta) = \frac{K}{\cos(\theta/k + \gamma_S)^k} , \quad k := \frac{2}{1 + \alpha} , \quad (18)$$

$$K := r(0) \cos^k \gamma_S = r(\theta_{\max}) \cos(\theta_{\max}/k + \gamma_S)^k \quad (19)$$

$$= \frac{R_{\max}}{\sin \theta_{\max}} \cos(\theta_{\max}/k + \gamma_S)^k . \quad (20)$$

Eq. (18) is a variant of the equations derived in [5]. As mentioned in [2], corrections are necessary to obtain uniform resolution images for large angles θ . In this case the shape of the mirror surface has to be calculated numerically.

Substituting $r(\theta)$ for the equi-angular mirror in Eqs. (13) and (14) we find

$$\lambda_{\theta}^*(\theta) = \frac{r(\theta)}{\alpha - 1} , \quad \lambda_{\varphi}^*(\theta) = \frac{r(\theta) \sin \theta}{\sin(\alpha\theta + 2\gamma_S) - \sin \theta} . \quad (21)$$

³ Exploiting the axial symmetry, it can be easily shown that $\frac{\partial \mathbf{x}(\theta, \varphi)}{\partial \theta}$ and $\frac{\partial \mathbf{x}(\theta, \varphi)}{\partial \varphi}$ are the principal curvature directions at point $\mathbf{x}(\theta, \varphi)$.

⁴ Interestingly, the differential equation $\lambda_{\theta}^*(\theta) = \lambda_{\varphi}^*(\theta)$ is solved by $r(\theta) = \frac{(1-b)r(0)}{1-b \cos \theta}$, $b \in \mathbb{R}$, which are conic sections.

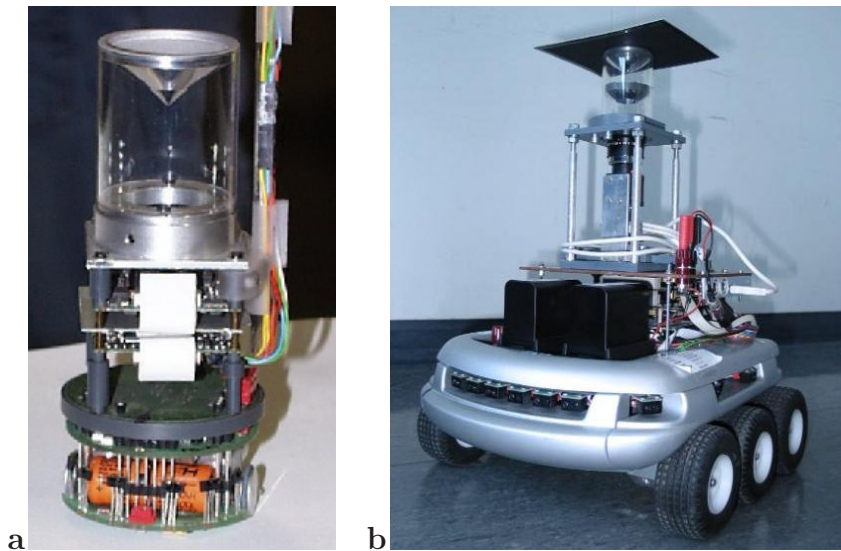


Fig. 2. Two mobile robots from K-Team SA (www.k-team.com) with omnidirectional stereo sensors used for indoor navigation. **a:** “Khepera”: diameter ≈ 5.5 cm, height ≈ 13 cm. **b:** “Koala”: size ≈ 35 cm \times 30 cm, height (including stereo sensor) ≈ 40 cm.

Since $\lambda_\varphi^*(0) = 0$ if $\gamma_S \neq 0$, $\mathbf{v}_\varphi(\theta, \varphi)$ touches the mirror surface at $\theta = 0$. If $\gamma_S = 0$, $\mathbf{v}_\varphi(\theta, \varphi)$ and $\mathbf{v}_\theta(\theta, \varphi)$ are in contact at $\theta = 0$, since $\lim_{\theta \rightarrow 0} \lambda_\varphi^*(\theta) = \lambda_\varphi^*(0)$. For small θ the imaging can then be described as reflexion at a sphere. In the case of a conical mirror ($\alpha = 1$, $\gamma_S > 0$) Eq. (21) give⁵

$$\lambda_\theta^*(\theta) \rightarrow \infty, \quad \lambda_\varphi^*(\theta) = \frac{r(\theta) \sin \theta}{\sin(\theta + 2\gamma_S) - \sin \theta}. \quad (22)$$

2.2 Examples

Our robots (see Fig. 2) are equipped with omnidirectional stereo sensors consisting of a single camera and equi-angular mirrors.

The mirror of the small “Khepera”-robot consists of two conical parts ($\alpha = 1$) with slightly different slopes. A more detailed description of the stereo sensor and its use for visual homing can be found in [6].

The sensor of the “Koala”-robot consists of two separate mirrors with the same diameter (46 mm) fixed inside a glass cylinder, see Fig. 3 a. A hole in the lower mirror permits imaging via the upper mirror. This configuration allows small dimensions and a comparatively large (vertical) stereo base line. In comparison to a similar panoramic stereo device suggested by Ollis et al. [2] we use the same angular magnification ($\alpha = 4$) for both mirrors which is achieved by set-

⁵ A planar mirror is described by $\alpha = 1$, $\gamma_S = 0 \rightsquigarrow \lambda_\theta^*, \lambda_\varphi^* \rightarrow \infty$.

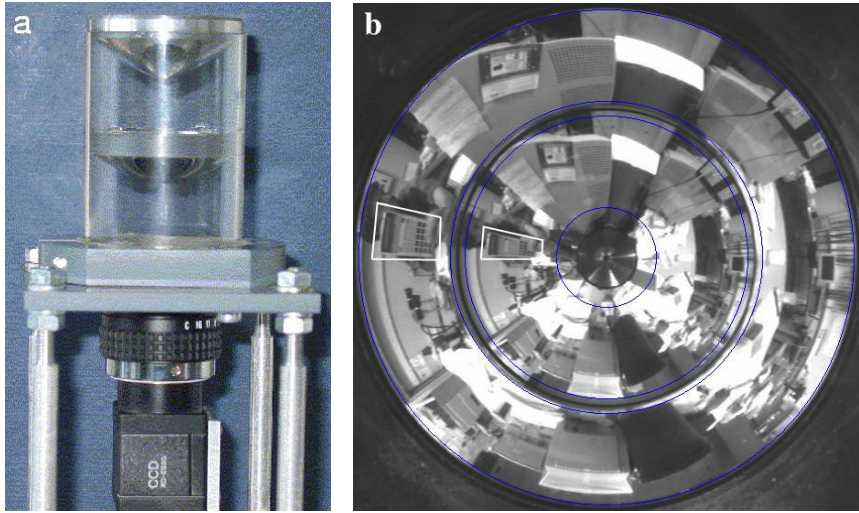


Fig. 3. a: Close-up view of Koala's stereo sensor. **b:** Stereo image, approx. size 550×550 pixel. The inner/outer part corresponds to the upper/lower mirror (in the image center one can see the reflection of the back side of the lower mirror). In the left side the stereo image of a pocket calculator held close to the sensor is highlighted by two polygons. The circles mark the areas that are used for stereo computation, see section 5 and Fig. 7.

ting $\gamma_S = 17.5^\circ$ for the upper mirror.⁶ A stereo image taken by the monochrome camera is shown in Fig. 3 b.

The virtual image surfaces,

$$\mathbf{v}_\theta(\theta, \varphi) = \frac{\alpha}{\alpha - 1} r(\theta) \mathbf{e}(\theta, \varphi) , \quad (23)$$

$$\mathbf{v}_\varphi(\theta, \varphi) = \frac{\sin(\alpha\theta + 2\gamma_S)}{\sin(\alpha\theta + 2\gamma_S) - \sin\theta} r(\theta) \mathbf{e}(\theta, \varphi) , \quad (24)$$

are depicted in Fig. 4 for both catadioptric stereo sensors.

For each mirror of the Koala sensor (Fig. 4 b) $\mathbf{v}_\theta(\theta, \varphi)$ and $\mathbf{v}_\varphi(\theta, \varphi)$ are comparatively close together and near to the reflective surface. Thus, when using a single mirror, the camera must be focused to a plane just behind the mirror surface to get a sharp image of distant objects. Furthermore, blur in the stereo image will be mainly due to the use of two mirrors with vertically separated virtual images. A good compromise is to focus the camera on a plane near the tip of the upper mirror.

⁶ Using the notation of Eqs. (18) and (20) the reflective surfaces of “configuration 5” in [2] can be described by $\gamma_S = 0$ (both mirrors) and $\alpha = 3$ (lower mirror), $\alpha = 7$ (upper mirror) resulting in different radial resolution in the inner and outer part of the stereo image.

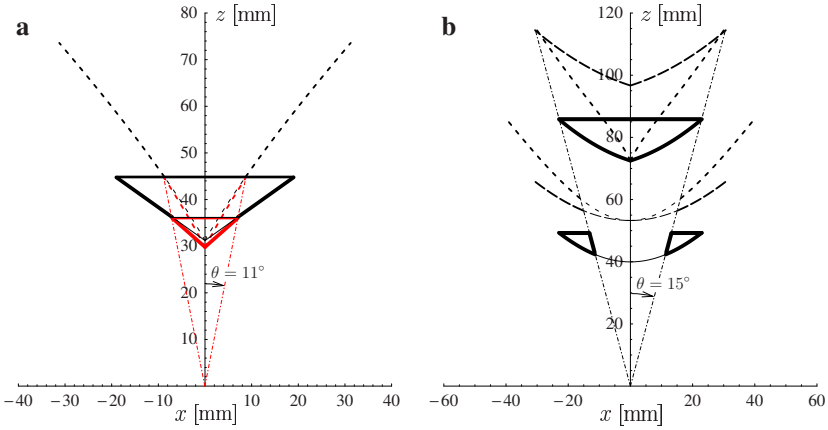


Fig. 4. Virtual image surfaces $\mathbf{v}_\theta(\theta, \varphi)$ (dashed) and $\mathbf{v}_\varphi(\theta, \varphi)$ (dotted) computed according to Eqs. (15), (16) and (21) for an axial plane. At some points these curves are continued by thin dashed and dotted curves, that do not contribute to the imaging (since the corresponding reflective surface does not exist). The mirrors are depicted by thick curves. Straight dash-dotted lines starting at the origin mark the θ -region of the inner mirror, i.e. the lower cone for the Khepera and the upper mirror for the Koala (note the different scaling). **a:** Conical stereo mirror of the Khepera. Since $\alpha = 1$, $\mathbf{v}_\theta(\theta, \varphi)$ lies “at infinity” for both mirror parts, see Eq. (22). **b:** Stereo sensor of the Koala: $\alpha = 4$, $R_{\max} = 23$ mm, $\gamma_S = 17.5^\circ$ (upper mirror), $\gamma_S = 0^\circ$ (lower mirror).

However, when using conical mirrors, the main reason for a blurred image will be the “infinite distance” between $\mathbf{v}_\theta(\theta, \varphi)$ and $\mathbf{v}_\varphi(\theta, \varphi)$ (Fig. 4 a).

3 Calculation of the Blur Region

As depicted in Fig. 5 a, the width of the blur region $\Delta\xi$ changes linearly with the distance d_0 of the image plane to the nodal point,

$$\frac{\Delta\xi}{D} = 1 - \frac{d_0}{d} . \quad (25)$$

D is the diameter of the lens aperture and d the image distance corresponding to a (virtual) point at distance z . The relation of z and d according to the thin lens law [7] is

$$z^{-1} + d^{-1} = f^{-1} . \quad (26)$$

Substituting (26), Eq. (25) can be transformed to

$$\frac{\Delta\xi}{D} = 1 - \left(\frac{d_0}{f} - \frac{d_0}{z} \right) = \frac{f(z_0 - z)}{z(z_0 - f)} , \quad (27)$$

where z_0 is the object plane to which the camera is focused.

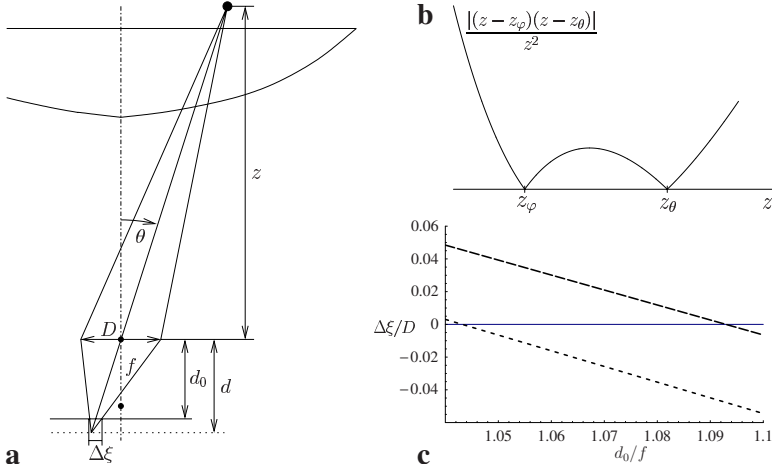


Fig. 5. a: Calculation of the blur region: If the image distance d which corresponds to an object at distance z deviates from the image plane at d_0 , then image blur will occur. The width of the defocused region $\Delta\xi$ depends linearly on the lens aperture D . **b:** Normalized area of the blur region, i.e. $A(z)D^{-2}f^{-2}z_\varphi z_\theta$, in dependence of distance z to the focused object plane. **c:** Calculated size of the blur regions $\Delta\xi_\theta^{\text{low}}$ (dashed) and $\Delta\xi_\varphi^{\text{up}}$ (dotted) for the Koala stereo sensor at $\theta = 15^\circ$ in dependence on the distance d_0 of the image plane, focal length is $f = 4.8$ mm.

To investigate Eq. (27), we assume that the camera is focused on a virtual image point of \mathbf{v}_θ , i.e. we set $z_0 = z_\theta := \mathbf{v}_\theta(\theta, \varphi) \mathbf{e}_z = \frac{\alpha}{\alpha-1} r(\theta) \cos \theta$. Considering the imaging of a point on \mathbf{v}_φ , i.e. $z = z_\varphi := \mathbf{v}_\varphi(\theta, \varphi) \mathbf{e}_z$, we obtain

$$\frac{\Delta\xi_\varphi}{D} = \frac{f(z_\theta - z_\varphi)}{z_\varphi(z_\theta - f)} \stackrel{z_\theta \gg f}{\approx} f \frac{z_\theta - z_\varphi}{z_\varphi z_\theta}. \quad (28)$$

Since $z_\theta, z_\varphi \propto r(\theta)$, see Eqs. (23) and (24) for $\alpha \neq 1$, we find $\Delta\xi_\varphi \propto \frac{Df}{r}$. For conical mirrors ($\alpha = 1, z_\theta \rightarrow \infty$) Eq. (28) is simplified to

$$\frac{\Delta\xi_\varphi}{D} = \frac{f}{z_\varphi} = \frac{f}{r(\theta)} \frac{\sin(\theta + 2\gamma_S) - \sin \theta}{\sin(\theta + 2\gamma_S) \cos \theta}. \quad (29)$$

The mirror shape will affect image quality especially for small dimensions of the catadioptric sensor. To get an idea of the size of the blur region, we set the parameters to typical values for the stereo sensor of the Khepera, e.g. $D \approx 1$ mm, $\frac{f}{z_\varphi} \approx \frac{1}{10}$, which results in $\Delta\xi_\varphi \approx 100 \mu\text{m}$ ($\Delta\xi_\theta \ll \Delta\xi_\varphi$ since the camera is assumed to be focused on \mathbf{v}_θ). By comparing this to the size of an element in the CCD-array of the camera, which is $\approx 10 \mu\text{m}$, it becomes apparent that such small conical mirrors cause significant blur.

For a single mirror the area of the blur region can be estimated by

$$A(z) \approx |\Delta\xi_\varphi(z)| |\Delta\xi_\theta(z)| \quad (30)$$

$$z \gg f \approx Df \frac{|z - z_\varphi|}{z_\varphi |z|} Df \frac{|z - z_\theta|}{z_\theta |z|} = \frac{D^2 f^2}{z_\varphi z_\theta} \frac{|z - z_\varphi| |z - z_\theta|}{z^2} . \quad (31)$$

An example of (31) with two minima at z_φ and z_θ is shown in Fig. 5 b. A similar curve was found by Baker and Nayar [1] for a hyperboloid mirror using numerical calculations.⁷

In standard applications one wants to find an image distance d_0 which ensures that both horizontal and vertical structures for all angles θ are imaged with little blur. In addition, two mirrors have to be taken into consideration for the Koala sensor. As can be seen from Fig. 4 b, the largest distance of the virtual image surfaces is at $\theta = 15^\circ$ between (for the lower mirror) $z_\theta^{\text{low}} = \mathbf{v}_\theta^{\text{low}} \mathbf{e}_z$ and (for the upper mirror) $z_\varphi^{\text{up}} = \mathbf{v}_\varphi^{\text{up}} \mathbf{e}_z$ (or z_θ^{up} which is almost identical to z_φ^{up} at $\theta = 15^\circ$). In Fig. 5 c, ξ_θ^{low} and ξ_φ^{up} are plotted against the distance d_0 of the image plane using Eq. (27). Both have the same absolute value of approx $0.02D$ at $d_0^* \approx 1.07f$. Since the focal length of the camera is $f = 4.8$ mm, d_0^* corresponds to an object distance $z_0 \approx 72.5$ – close to the tip of the upper cone. If we assume $f/D = 5.6$ which is a standard value for indoor imaging, the expected maximum blur region $\Delta\xi$ is of the magnitude of one or two elements of the CCD-array.⁸ This is significantly better than for the stereo sensor of the Khepera but still restricts the use of high resolution cameras.

4 Caustics of the Stereo Sensors

In this section the caustic of a catadioptric system with equi-angular magnification factor is calculated. The caustic is the curve to which the reflected rays of light are tangents. Because of axial symmetry we omit the parameter φ in the following. As described in [8] a point on the caustic lies on the straight line of the light ray given by

$$\mathbf{l}(\theta, \lambda) = \mathbf{x}(\theta) + \lambda \mathbf{e}_o(\theta) = \mathbf{x}(\theta) + \lambda \mathbf{e}(\Theta_o(\theta)) , \quad \lambda \in \mathbb{R} . \quad (32)$$

The caustic is then defined by $\mathbf{c}(\theta) := \mathbf{l}(\theta, \lambda(\theta))$, where $\lambda(\theta)$ is the solution of the equation denoted in [8],

$$\det\left(\frac{\partial}{\partial\theta}\mathbf{l}(\theta, \lambda), \frac{\partial}{\partial\lambda}\mathbf{l}(\theta, \lambda)\right) = 0 . \quad (33)$$

The solution of (33) after substitution of (32) is

$$\lambda(\theta) = -\frac{1}{\Theta_o'(\theta)} \left(r(\theta) \cos(\theta - \Theta_o(\theta)) + r'(\theta) \sin(\theta - \Theta_o(\theta)) \right) . \quad (34)$$

⁷ The small offsets at the minima mentioned in [1] are probably due to higher order terms which were omitted in our calculations.

⁸ If the inner and outer rings of the stereo image are unwarped to the same size, a small blur in the outer image part (which corresponds to the lower mirror) will have less effect.

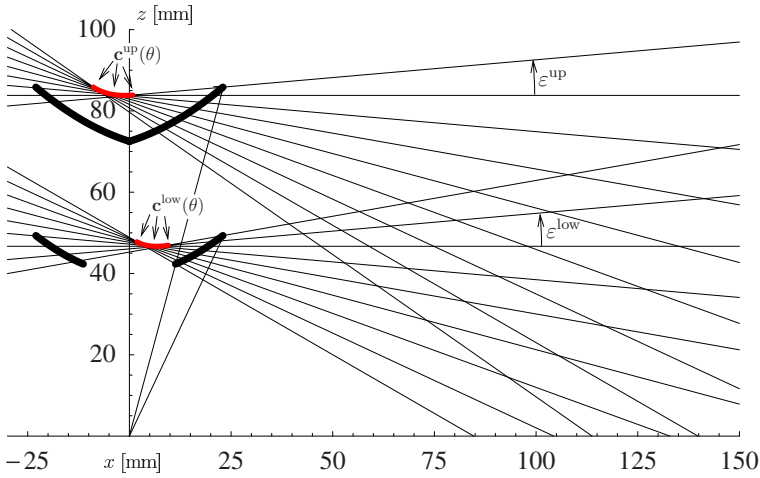


Fig. 6. Rays and caustic curves (marked by arrows) of the Koala stereo sensor. The reflective surfaces are depicted by thick curves. The rays are plotted for $\theta = 1.25^\circ l$, $l = 0, 1, \dots, 12$ for the upper mirror and $\theta = 15^\circ + 1.25^\circ l$, $l = 0, 1, \dots, 8$ for the lower mirror. The corresponding (used) angles of elevation $\varepsilon = \frac{\pi}{2} - \theta_o$ are approx. $\varepsilon^{\text{up}} \in [-35^\circ, +5^\circ]$ and $\varepsilon^{\text{low}} \in [-30^\circ, +10^\circ]$ respectively.

For the equi-angular mirror, for which $r(\theta)$ and $\Theta_o(\theta)$ are given by Eqs. (17) and (18), (34) yields $\lambda(\theta) = -\frac{r(\theta)}{\alpha}$ and the caustic (in the x - z -plane) is given by

$$\mathbf{c}(\theta) = \frac{K}{\cos(\theta/k + \gamma_S)^k} \left[\begin{pmatrix} \sin \theta \\ \cos \theta \end{pmatrix} - \frac{1}{\alpha} \begin{pmatrix} \sin(\alpha\theta + 2\gamma_S) \\ -\cos(\alpha\theta + 2\gamma_S) \end{pmatrix} \right]. \quad (35)$$

The caustics of the Koala stereo sensor are shown in Fig. 6. Since the vertical extension of the caustics is small, the stereo imaging can be described in good approximation by means of two virtual points close to the caustics. The distance of all rays in the relevant angular domains, i.e. $\varepsilon^{\text{up}} \in [-35^\circ, +5^\circ]$ and $\varepsilon^{\text{low}} \in [-30^\circ, +10^\circ]$ to the points $\mathbf{p}^{\text{up}} \approx (-5.1 \text{ mm}, 83.8 \text{ mm})$ and $\mathbf{p}^{\text{low}} \approx (4.9 \text{ mm}, 46.5 \text{ mm})$ respectively is below 0.4 mm. In 3D these points form two vertically separated circles rather close to the symmetry axis. Thus, the vertical baseline of the stereo system is $\approx 37 \text{ mm}$.

By substituting $\alpha = 1$ into (35), we see that the caustic of a simple conical mirror is independent of θ , i.e. the caustic surface consists of a single point in 2D, $\mathbf{c}(\theta|\alpha = 1) = 2K(-\sin \gamma_S, \cos \gamma_S)^\top$, or a ring in 3D. This fact has already been described before, e.g. in [9], [6]. For the stereo sensor of the Khepera, these two points have the coordinates $(-29.8 \text{ mm}, 33.7 \text{ mm})$ and $(-29.5 \text{ mm}, 41.4 \text{ mm})$. Therefore, the two circles have approximately the same radius.

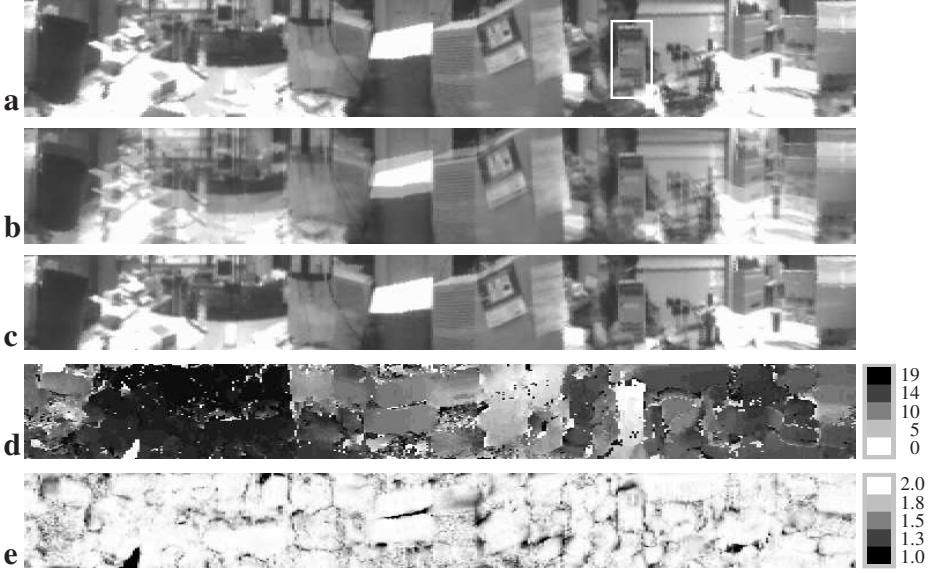


Fig. 7. “Physiological” disparity computation: **a:** Unwarped image $I^{\text{up}}(x, y)$ (corresponding to the upper mirror; size 360×100 pixel) of the stereo image shown in Fig. 3 b. The rectangle highlights the pocket calculator near the sensor. **b:** Superposition of the unwarped upper and lower images. **c:** Fused image, $\frac{1}{2}(I^{\text{up}}(x, y) + I^{\text{low}}(x, y + d(x, y)))$, using the disparity map $d(x, y)$. **d:** Computed disparity map, $d(x, y)$ is the preferred disparity of the locally most active neuron. The “calculator area” contains small disparities. **e:** Activity map $a(x, y) \in [0, 2]$ of the locally most active neurons. An activity well below 1.5 indicates that the corresponding disparity is uncertain. This occurs primarily at occlusion regions or low-textured image parts.

5 Disparity Computation with the Koala Stereo Sensor

After unwarping the stereo image we use a stereo algorithm based on the binocular energy model [10], which can explain most of the experimental data on disparity-tuned complex cells in the visual cortex of cats and monkeys. The implementation with an additional normalization stage is similar to that described in [11]. Two (spatial) frequency channels (center frequencies are $\nu_1 = 7.5 \nu_0$ and $\nu_2 = 25 \nu_0$, $\nu_0 = (100 [\text{pixel}])^{-1}$) with a bandwidth of two octaves are used. An example of the unwarped images and the resulting disparity map for the stereo image of Fig. 3 b is shown in Fig. 7.

6 Conclusions

We have presented an analytical description for the virtual image zones of axially symmetrical catadioptric sensors that allows a quantitative evaluation of the imaging quality of the system. An extension to non-axially symmetric reflective

surfaces should be straight forward. If the surfaces for horizontally and vertically parallel incoming light rays, \mathbf{v}_φ and \mathbf{v}_θ , are widely separated (compared to distance from the nodal point to \mathbf{v}_φ and \mathbf{v}_θ , see Eq. (28)), significant blur will occur especially in indoor environments where the lens aperture must be opened widely. We showed that small conical mirrors have a comparatively low imaging quality.

We also presented a panoramic stereo sensor consisting of a single camera directed towards two vertically separated reflective surfaces with the same constant angular magnification. Because of its compact size it is well suited for omnidirectional disparity/depth estimation on mobile robots for localization and obstacle avoidance purposes.

References

1. Baker, S., Nayar, S.: A theory of catdioptric image formation. In: ICCV 1998, IEEE Computer Society (1998) 35–40
2. Ollis, M., Herman, H., Singh, S.: Analysis and design of panoramic stereo vision using equi-angular pixel cameras. Technical report, CMU (1999)
3. <http://www.cis.upenn.edu/~kostas/omni.html>: The page of omnidirectional vision (2003)
4. Oren, M., Nayar, S.: A theory of specular surface geometry. Int. J. Computer Vision **24** (1997) 105–124
5. Chahl, J.S., Srinivasan, M.V.: Reflective surfaces for panoramic imaging. Applied Optics **36** (1997) 8275–8285
6. Stürzl, W., Mallot, H.: Vision-based homing with a panoramic stereo sensor. In: BMCV 2002. Volume 2525 of LNCS. (2002) 620–628
7. Hecht, E.: Optik. 2. edn. Addison-Wesley (1992)
8. Swaminathan, R., Grossberg, M., Nayar, S.: Caustics of catadioptric cameras. In: ICCV 2001. (2001) 2–9
9. Spacek, L.: Omnidirectional catadioptric vision sensor with conical mirrors. In: TIMR 2003, Bristol, UK (2003)
10. Ohzawa, I., DeAngelis, G., Freeman, R.: Stereoscopic depth discrimination in the visual cortex: Neurons ideally suited as disparity detectors. Science **249** (1990) 1037–1041
11. Stürzl, W., Hoffmann, U., Mallot, H.: Vergence control and disparity estimation with energy neurons: Theory and implementation. In: ICANN 2002. Volume 2415 of LNCS. (2002) 1255–1260

A Appendix: Derivation of Eq. (11)

Taylor expansion of $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{n}}^0$ about (θ, φ) yields

$$\tilde{\mathbf{x}} := \mathbf{x}(\theta + \Delta\theta, \varphi + \Delta\varphi) \approx \mathbf{x} + \Delta\mathbf{x} , \quad (36)$$

$$\tilde{\mathbf{n}}^0 := \mathbf{n}^0(\theta + \Delta\theta, \varphi + \Delta\varphi) \approx \mathbf{n}^0 + \Delta\mathbf{n} + \Delta^2\mathbf{n} , \quad (37)$$

$$\Delta\mathbf{x} := \frac{\partial\mathbf{x}}{\partial\theta}\Delta\theta + \frac{\partial\mathbf{x}}{\partial\varphi}\Delta\varphi , \quad \Delta\mathbf{n} := \frac{\partial\mathbf{n}^0}{\partial\theta}\Delta\theta + \frac{\partial\mathbf{n}^0}{\partial\varphi}\Delta\varphi , \quad (38)$$

$$\Delta^2\mathbf{n} := \frac{1}{2}\frac{\partial^2\mathbf{n}^0}{\partial\theta^2}\Delta\theta^2 + \frac{\partial^2\mathbf{n}^0}{\partial\varphi\partial\theta}\Delta\theta\Delta\varphi + \frac{1}{2}\frac{\partial^2\mathbf{n}^0}{\partial\varphi^2}\Delta\varphi^2 . \quad (39)$$

Using (37), $\tilde{\mathbf{e}}$ can be approximated up to second-order by

$$\tilde{\mathbf{e}} = \mathbf{e}_o - 2(\tilde{\mathbf{n}}^0 \mathbf{e}_o) \tilde{\mathbf{n}}^0 \approx \mathbf{e} + \Delta \mathbf{e} + \Delta^2 \mathbf{e} , \quad (40)$$

$$\Delta \mathbf{e} := -2[(\Delta \mathbf{n} \mathbf{e}_o) \mathbf{n}^0 + (\mathbf{n}^0 \mathbf{e}_o) \Delta \mathbf{n}] , \quad (41)$$

$$\Delta^2 \mathbf{e} := -2[(\Delta^2 \mathbf{n} \mathbf{e}_o) \mathbf{n}^0 + (\Delta \mathbf{n} \mathbf{e}_o) \Delta \mathbf{n} + (\mathbf{n}^0 \mathbf{e}_o) \Delta^2 \mathbf{n}] . \quad (42)$$

From the Taylor expansion up to second-order of the numerator and the denominator,

$$(\tilde{\mathbf{x}} - \mathbf{x})((\mathbf{e}\tilde{\mathbf{e}})\tilde{\mathbf{e}} - \mathbf{e}) \approx (\Delta \mathbf{x} \Delta \mathbf{e}) + (\mathbf{e} \Delta \mathbf{x})(\mathbf{e} \Delta \mathbf{e}) , \quad (43)$$

$$(\mathbf{e}\tilde{\mathbf{e}})^2 - 1 \approx 2(\mathbf{e} \Delta \mathbf{e}) + 2(\mathbf{e} \Delta^2 \mathbf{e}) + (\mathbf{e} \Delta \mathbf{e})^2 , \quad (44)$$

Eq. (2) can be approximated by

$$\lambda^* \approx \frac{(\Delta \mathbf{x} \Delta \mathbf{e}) + (\mathbf{e} \Delta \mathbf{x})(\mathbf{e} \Delta \mathbf{e})}{2(\mathbf{e} \Delta \mathbf{e}) + 2(\mathbf{e} \Delta^2 \mathbf{e}) + (\mathbf{e} \Delta \mathbf{e})^2} . \quad (45)$$

Substituting $\mathbf{e}_o = \mathbf{e} - 2(\mathbf{n}^0 \mathbf{e}) \mathbf{n}^0$ into (41) and exploiting the fact that $\mathbf{n}^0 \Delta \mathbf{n} = 0$ (if the reflective surface is given by $\mathbf{x}(\theta, \varphi) = r(\theta) \mathbf{e}(\theta, \varphi)$, i.e. an axially symmetrical mirror), straight forward calculation yields $\mathbf{e} \Delta \mathbf{e} = 0$. Thus, Eq. (45) is simplified to

$$\lambda^* \approx \frac{1}{2} \frac{\Delta \mathbf{x} \Delta \mathbf{e}}{\mathbf{e} \Delta^2 \mathbf{e}} . \quad (46)$$

By means of $\mathbf{n} \Delta \mathbf{x} = 0$ and $\frac{\partial \mathbf{x}}{\partial \theta} \frac{\partial \mathbf{n}^0}{\partial \varphi} = \frac{\partial \mathbf{x}}{\partial \varphi} \frac{\partial \mathbf{n}^0}{\partial \theta} = 0$ we obtain

$$\Delta \mathbf{x} \Delta \mathbf{e} = 2(\mathbf{n}^0 \mathbf{e})(\Delta \mathbf{n} \Delta \mathbf{x}) \quad (47)$$

$$= 2(\mathbf{n}^0 \mathbf{e}) \left[\left(\frac{\partial \mathbf{x}}{\partial \theta} \frac{\partial \mathbf{n}^0}{\partial \theta} \right) \Delta \theta^2 + \left(\frac{\partial \mathbf{x}}{\partial \varphi} \frac{\partial \mathbf{n}^0}{\partial \varphi} \right) \Delta \varphi^2 \right] . \quad (48)$$

Using $\mathbf{n}^0 \Delta \mathbf{n} = 0$, $\frac{\partial^2 \mathbf{n}^0}{\partial \varphi \partial \theta} \mathbf{n}^0 = 0$ and $\frac{\partial \mathbf{n}^0}{\partial \varphi} \mathbf{e} = 0$, simple calculus leads to

$$\mathbf{e} \Delta^2 \mathbf{e} = 4(\mathbf{n}^0 \mathbf{e})^2 (\mathbf{n}^0 \Delta^2 \mathbf{n}) - 2(\Delta \mathbf{n} \mathbf{e})^2 \quad (49)$$

$$= 2(\mathbf{n}^0 \mathbf{e})^2 \left[\left(\mathbf{n}^0 \frac{\partial^2 \mathbf{n}^0}{\partial \theta^2} \right) \Delta \theta^2 + \left(\mathbf{n}^0 \frac{\partial^2 \mathbf{n}^0}{\partial \varphi^2} \right) \Delta \varphi^2 \right] - 2 \left(\frac{\partial \mathbf{n}^0}{\partial \theta} \mathbf{e} \right)^2 \Delta \theta^2 . \quad (50)$$

Substituting

$$\mathbf{n}^0 \mathbf{e} = - \left[1 + \left(\frac{r'(\theta)}{r(\theta)} \right)^2 \right]^{-\frac{1}{2}} , \quad (51)$$

$$\frac{\partial \mathbf{x}}{\partial \theta} \frac{\partial \mathbf{n}^0}{\partial \theta} = r(\theta) (\mathbf{n}^0 \mathbf{e}) \left[1 + 2 \left(\frac{r'(\theta)}{r(\theta)} \right)^2 - \frac{r''(\theta)}{r(\theta)} \right] , \quad (52)$$

$$\frac{\partial \mathbf{x}}{\partial \varphi} \frac{\partial \mathbf{n}^0}{\partial \varphi} = r(\theta) \sin \theta (\mathbf{n}^0 \mathbf{e}) \left[\sin \theta - \cos \theta \frac{r'(\theta)}{r(\theta)} \right] , \quad (53)$$

$$\mathbf{n}^0 \frac{\partial^2 \mathbf{n}^0}{\partial \theta^2} = -(\mathbf{n}^0 \mathbf{e})^4 \left[1 + 2 \left(\frac{r'(\theta)}{r(\theta)} \right)^2 - \frac{r''(\theta)}{r(\theta)} \right]^2 , \quad (54)$$

$$\mathbf{n}^0 \frac{\partial^2 \mathbf{n}^0}{\partial \varphi^2} = -(\mathbf{n}^0 \mathbf{e})^2 \left[\sin \theta - \cos \theta \frac{r'(\theta)}{r(\theta)} \right]^2, \quad (55)$$

$$\frac{\partial \mathbf{n}^0}{\partial \theta} \mathbf{e} = (\mathbf{n}^0 \mathbf{e})^3 \frac{r'(\theta)}{r(\theta)} \left[1 + 2 \left(\frac{r'(\theta)}{r(\theta)} \right)^2 - \frac{r''(\theta)}{r(\theta)} \right] \quad (56)$$

into (48) and (50) we finally obtain from (46)

$$\begin{aligned} \lambda^* \approx & -\frac{1}{2}r \left[1 + \left(\frac{r'}{r} \right)^2 \right] \\ & \times \frac{\Delta \theta^2 \left[1 + 2 \left(\frac{r'}{r} \right)^2 - \frac{r''}{r} \right] + \Delta \varphi^2 \sin \theta \left[\sin \theta - \cos \theta \frac{r'}{r} \right]}{\Delta \theta^2 \left[1 + 2 \left(\frac{r'}{r} \right)^2 - \frac{r''}{r} \right]^2 + \Delta \varphi^2 \left[\sin \theta - \cos \theta \frac{r'}{r} \right]^2}. \end{aligned} \quad (57)$$

Author Index

- Abraham, Isabelle IV-37
Agarwal, Ankur III-54
Agarwal, Sameer II-483
Ahmadyfard, Ali R. IV-342
Ahonen, Timo I-469
Ahuja, Narendra I-508, IV-602
Aloimonos, Yiannis IV-229
Antone, Matthew II-262
Argyros, Antonis A. III-368
Armospach, Jean-Paul III-546
Arnaud, Elise III-302
Arora, Himanshu I-508
Åström, Kalle III-252
Attias, Hagai IV-546
Aubert, Gilles IV-1
Auer, P. II-71
Avidan, Shai IV-428
Avraham, Tamar II-58
Ayache, Nicholas III-79
- Bab-Hadiashar, Alireza I-83
Baker, Patrick IV-229
Balch, Tucker IV-279
Bar, Leah II-166
Barnard, Kobus I-350
Bart, Evgeniy II-152
Bartoli, Adrien II-28
Basalamah, Saleh III-417
Basri, Ronen I-574, II-99
Bayerl, Pierre III-158
Bayro-Corrochano, Eduardo I-536
Bebis, George IV-456
Bect, Julien IV-1
Belhumeur, Peter I-146
Belongie, Serge II-483, III-170
Bennamoun, Mohammed II-495
Besserer, Bernard III-264
Bharath, Anil A. I-482, III-417
Bicego, Manuele II-202
Bille, Philip II-313
Bissacco, Alessandro III-456
Blake, Andrew I-428, II-391
Blanc-Féraud, Laure IV-1
Borenstein, Eran III-315
Bouthemy, Patrick III-145
Bowden, Richard I-390
- Brady, Michael I-228, I-390
Brand, Matthew II-262
Bretzner, Lars I-322
Bronstein, Alexander M. II-225
Bronstein, Michael M. II-225
Brostow, Gabriel J. III-66
Broszio, Hellward I-523
Brown, M. I-428
Brox, Thomas II-578, IV-25
Bruckstein, Alfred M. III-119
Bruhn, Andrés IV-25, IV-205
Bülow, Thomas III-224
Burger, Martin I-257
Burgeth, Bernhard IV-155
Byvatov, Evgeny II-152
- Calway, Andrew II-379
Caputo, Barbara IV-253
Carbonetto, Peter I-350, III-1
Carlsson, Stefan II-518, IV-442
Chai, Jin-xiang IV-573
Chambolle, Antonin IV-1
Charbonnier, Pierre II-341
Charnoz, Arnaud IV-267
Chellappa, Rama I-588
Chen, Chu-Song I-108, II-190
Chen, Jiun-Hung I-108
Chen, Min III-468
Chen, Qian III-521
Chen, Yu-Ting II-190
Cheng, Qiansheng I-121
Chiuso, Alessandro III-456
Christoudias, Chris Mario IV-481
Chua, Chin-Seng III-288
Chung, Albert C.S. II-353
Cipolla, Roberto II-391
Claus, David IV-469
Cohen, Isaac II-126
Cohen, Michael II-238
Comaniciu, Dorin I-336, I-549
Cootes, T.F. IV-316
Coquerelle, Mathieu II-28
Cremers, Daniel IV-74
Cristani, Marco II-202
Cristóbal, Gabriel III-158

- Dahmen, Hansjürgen I-614
 Dalal, Navneet I-549
 Daniilidis, Kostas II-542
 Darcourt, Jacques IV-267
 Darrell, Trevor IV-481, IV-507
 Davis, Larry S. I-175, III-482
 Dellaert, Frank III-329, IV-279
 Demirci, M. Fatih I-322
 Demirdjian, David III-183
 Deriche, Rachid II-506, IV-127
 Derpanis, Konstantinos G. I-282
 Devernay, Frédéric I-495
 Dewaele, Guillaume I-495
 Dickinson, Sven I-322
 Doretto, Gianfranco II-591
 Dovgard, Roman II-99
 Drew, Mark S. III-582
 Drummond, Tom II-566
 Duan, Ye III-238
 Duin, Robert P.W. I-562
 Dunagan, B. IV-507
 Duraiswami, Ramani III-482

 Ebner, Marc III-276
 Eklundh, Jan-Olof IV-253, IV-366
 Engbers, Erik A. III-392
 Eong, Kah-Guan Au II-139
 Eriksson, Martin IV-442
 Essa, Irfan III-66

 Fagerström, Daniel IV-494
 Faugeras, Olivier II-506, IV-127, IV-141
 Favaro, Paolo I-257
 Feddern, Christian IV-155
 Fei, Huang III-497
 Fergus, Robert I-242
 Fermüller, Cornelia III-405
 Ferrari, Vittorio I-40
 Finlayson, Graham D. III-582
 Fischer, Sylvain III-158
 Fitzgibbon, Andrew W. IV-469
 Freitas, Nando de I-28, I-350, III-1
 Freixenet, Jordi II-250
 Fritz, Mario IV-253
 Frolova, Darya I-574
 Frome, Andrea III-224
 Fua, Pascal II-405, II-566, III-92
 Fuh, Chiou-Shann I-402
 Furukawa, Yasutaka II-287
 Fussenegger, M. II-71

 Gavril, Darin M. IV-241
 Gheissari, Niloofar I-83
 Ghodsi, Ali IV-519
 Giblin, Peter II-313, II-530
 Giebel, Jan IV-241
 Ginneken, Bram van I-562
 Goldlücke, Bastian II-366
 Gool, Luc Van I-40
 Grossauer, Harald II-214
 Gumerov, Nail III-482
 Gupta, Rakesh I-215
 Guskov, Igor I-133
 Gyaourova, Aglika IV-456

 Hadid, Abdenour I-469
 Haider, Christoph IV-560
 Hanbury, Allan IV-560
 Hancock, Edwin R. III-13, IV-114
 Hartley, Richard I. I-363
 Hayman, Eric IV-253
 Heinrich, Christian III-546
 Heitz, Fabrice III-546
 Herda, Lorna II-405
 Hershey, John IV-546
 Hertzmann, Aaron II-299, II-457
 Hidović, Džena IV-414
 Ho, Jeffrey I-456
 Ho, Purdy III-430
 Hoey, Jesse III-26
 Hofer, Michael I-297, IV-560
 Hong, Byung-Woo IV-87
 Hong, Wei III-533
 Horaud, Radu I-495
 Hsu, Wynne II-139
 Hu, Yuxiao I-121
 Hu, Zhanyi I-190, I-442
 Huang, Fay II-190
 Huang, Jiayuan IV-519
 Huber, Daniel III-224

 Ieng, Sio-Song II-341
 Ikeda, Sei II-326
 Irani, Michal II-434, IV-328

 Jacobs, David W. I-588, IV-217
 Jawahar, C.V. IV-168
 Je, Changsoo I-95
 Ji, Hui III-405
 Jia, Jiaya III-342
 Jin, Hailin II-114

- Jin, Jesse S. I-270
 Johansen, P. IV-180
 Jones, Eagle II-591
 Joshi, Shantanu III-570

 Kadir, Timor I-228, I-390
 Kaess, Michael III-329
 Kanade, Takeo III-558, IV-573
 Kanatani, Kenichi I-310
 Kang, Sing Bing II-274
 Kasturi, Rangachar IV-390
 Kervrann, Charles III-132
 Keselman, Yakov I-322
 Khan, Zia IV-279
 Kimia, Benjamin II-530
 Kimmel, Ron II-225
 Kiryati, Nahum II-166, IV-50
 Kittler, Josef IV-342
 Kohlberger, Timo IV-205
 Kokkinos, Iasonas II-506
 Kolluri, Ravi III-224
 Koulibaly, Pierre Malick IV-267
 Koudelka, Melissa I-146
 Kriegman, David I-456, II-287, II-483
 Krishnan, Arun I-549
 Krishnan, Sriram I-336
 Kristjansson, Trausti IV-546
 Kück, Hendrik III-1
 Kuijper, Arjan II-313
 Kumar, Pankaj I-376
 Kumar, R. III-442
 Kuthirummal, Sujit IV-168
 Kwatra, Vivek III-66
 Kwolek, Bogdan IV-192

 Lagrange, Jean Michel IV-37
 Lee, Kuang-chih I-456
 Lee, Mong Li II-139
 Lee, Mun Wai II-126
 Lee, Sang Wook I-95
 Lenglet, Christophe IV-127
 Leung, Thomas I-203
 Levin, Anat I-602, IV-377
 Lhuillier, Maxime I-163
 Lim, Jongwoo I-456, II-470
 Lim, Joo-Hwee I-270
 Lin, Stephen II-274
 Lin, Yen-Yu I-402
 Lindenbaum, Michael II-58, III-392, IV-217

 Lingrand, Diane IV-267
 Little, James J. I-28, III-26
 Liu, Ce II-603
 Liu, Tyng-Luh I-402
 Liu, Xiuwen III-570, IV-62
 Lladó, Xavier II-250
 Loog, Marco I-562, IV-14
 López-Franco, Carlos I-536
 Lourakis, Manolis I.A. III-368
 Lowe, David G. I-28
 Loy, Gareth IV-442
 Lu, Cheng III-582

 Ma, Yi I-1, III-533
 Magnor, Marcus II-366
 Maire, Michael I-55
 Malik, Jitendra III-224
 Mallick, Satya P. II-483
 Mallot, Hanspeter A. I-614
 Manay, Siddharth IV-87
 Manduchi, Roberto IV-402
 Maragos, Petros II-506
 Marsland, S. IV-316
 Martí, Joan II-250
 Matei, B. III-442
 Matsushita, Yasuyuki II-274
 Maurer, Jr., Calvin R. III-596
 McKenna, Stephen J. IV-291
 McMillan, Leonard II-14
 McRobbie, Donald III-417
 Medioni, Gérard IV-588
 Meltzer, Jason I-215
 Mémin, Etienne III-302
 Mendonça, Paulo R.S. II-554
 Mian, Ajmal S. II-495
 Mikolajczyk, Krystian I-69
 Miller, James II-554
 Mio, Washington III-570, IV-62
 Mittal, Anurag I-175
 Montagnat, Johan IV-267
 Mordohai, Philippos IV-588
 Moreels, Pierre I-55
 Morency, Louis-Philippe IV-481
 Moreno, Pedro III-430
 Moses, Yael IV-428
 Moses, Yoram IV-428
 Muñoz, Xavier II-250
 Murino, Vittorio II-202

 Narayanan, P.J. IV-168
 Nechyba, Michael C. II-178

- Neumann, Heiko III-158
 Ng, Jeffrey I-482
 Nguyen, Hieu T. II-446
 Nicolau, Stéphane III-79
 Nielsen, Mads II-313, IV-180
 Nillius, Peter IV-366
 Nir, Tal III-119
 Nistér, David II-41
 Noblet, Vincent III-546
- Odehnal, Boris I-297
 Okada, Kazunori I-549
 Okuma, Kenji I-28
 Oliensis, John IV-531
 Olsen, Ole Fogh II-313
 Opelt, A. II-71
 Osadchy, Margarita IV-217
 Owens, Robyn II-495
- Padfield, Dirk II-554
 Pallawala, P.M.D.S. II-139
 Papenberg, Nils IV-25
 Paris, Sylvain I-163
 Park, JinHyeong IV-390
 Park, Rae-Hong I-95
 Pavlidis, Ioannis IV-456
 Peleg, Shmuel IV-377
 Pelillo, Marcello IV-414
 Pennec, Xavier III-79
 Perez, Patrick I-428
 Perona, Pietro I-55, I-242, III-468
 Petrović, Vladimir III-380
 Pietikäinen, Matti I-469
 Pinz, A. II-71
 Piriou, Gwenaëlle III-145
 Pollefeys, Marc III-509
 Pollitt, Anthony II-530
 Ponce, Jean II-287
 Pottmann, Helmut I-297, IV-560
 Prados, Emmanuel IV-141
- Qin, Hong III-238
 Qiu, Huaijun IV-114
 Quan, Long I-163
- Rahimi, A. IV-507
 Ramalingam, Sri Kumar II-1
 Ramamoorthi, Ravi I-146
 Ranganath, Surendra I-376
 Redondo, Rafael III-158
 Reid, Ian III-497
- Ricketts, Ian W. IV-291
 Riklin-Raviv, Tammy IV-50
 Roberts, Timothy J. IV-291
 Rohlfing, Torsten III-596
 Rosenhahn, Bodo I-414
 Ross, David II-470
 Rother, Carsten I-428
 Russakoff, Daniel B. III-596
- Saisan, Payam III-456
 Samaras, Dimitris III-238
 Sarel, Bernard IV-328
 Sato, Tomokazu II-326
 Satoh, Shin'ichi III-210
 Savarese, Silvio III-468
 Sawhney, H.S. III-442
 Schaffalitzky, Frederik I-363, II-41, II-85
 Schmid, Cordelia I-69
 Schnörr, Christoph IV-74, IV-205, IV-241
 Schuurmans, Dale IV-519
 Seitz, Steven M. II-457
 Sengupta, Kuntal I-376
 Sethi, Amit II-287
 Shahrokni, Ali II-566
 Shan, Y. III-442
 Shashua, Amnon III-39
 Shokoufandeh, Ali I-322
 Shum, Heung-Yeung II-274, II-603, III-342
 Simakov, Denis I-574
 Singh, Maneesh I-508
 Sinha, Sudipta III-509
 Sivic, Josef II-85
 Smeulders, Arnold W.M. II-446, III-392
 Smith, K. IV-316
 Soatto, Stefano I-215, I-257, II-114, II-591, III-456, IV-87
 Sochen, Nir II-166, IV-50, IV-74
 Soler, Luc III-79
 Sommer, Gerald I-414
 Sorgi, Lorenzo II-542
 Spacek, Libor IV-354
 Spira, Alon II-225
 Srivastava, Anuj III-570, IV-62
 Steedly, Drew III-66
 Steiner, Tibor IV-560
 Stewenius, Henrik III-252
 Stürzl, Wolfgang I-614
 Sturm, Peter II-1, II-28

- Sugaya, Yasuyuki I-310
 Sullivan, Josephine IV-442
 Sun, Jian III-342
 Suter, David III-107
 Szepesvári, Csaba I-16

 Taleghani, Ali I-28
 Tang, Chi-Keung II-419, III-342
 Tarel, Jean-Philippe II-341
 Taylor, C.J. IV-316
 Teller, Seth II-262
 Thiesson, Bo II-238
 Thiré, Cedric III-264
 Thormählen, Thorsten I-523
 Thureson, Johan II-518
 Todorovic, Sinisa II-178
 Tomasi, Carlo III-596
 Torma, Péter I-16
 Torr, Philip I-428
 Torresani, Lorenzo II-299
 Torsello, Andrea III-13, IV-414
 Treuille, Adrien II-457
 Triggs, Bill III-54, IV-100
 Tsin, Yanghai III-558
 Tsotsos, John K. I-282
 Tu, Zhuowen III-195
 Turek, Matt II-554
 Tuytelaars, Tinne I-40
 Twining, C.J. IV-316

 Ullman, Shimon II-152, III-315
 Urtasun, Raquel II-405, III-92

 Vasconcelos, Nuno III-430
 Vemuri, Baba C. IV-304
 Vidal, René I-1

 Wada, Toshikazu III-521
 Wallner, Johannes I-297
 Wang, Hanzi III-107
 Wang, Jue II-238
 Wang, Zhizhou IV-304
 Weber, Martin II-391
 Weickert, Joachim II-578, IV-25, IV-155, IV-205
 Weimin, Huang I-376
 Weiss, Yair I-602, IV-377
 Weissenfeld, Axel I-523

 Welk, Martin IV-155
 Wen, Fang II-603
 Wildes, Richard P. I-282
 Wills, Josh III-170
 Windridge, David I-390
 Wolf, Lior III-39
 Wong, Wilbur C.K. II-353
 Wu, Fuchao I-190
 Wu, Haiyuan III-521
 Wu, Tai-Pang II-419
 Wu, Yihong I-190

 Xiao, Jing IV-573
 Xu, Ning IV-602
 Xu, Yingqing II-238
 Xydeas, Costas III-380

 Yan, Shuicheng I-121
 Yang, Liu III-238
 Yang, Ming-Hsuan I-215, I-456, II-470
 Yao, Annie II-379
 Yao, Jian-Feng III-145
 Yezzi, Anthony J. II-114, IV-87
 Ying, Xianghua I-442
 Yokoya, Naokazu II-326
 Yu, Hongchuan III-288
 Yu, Jingyi II-14
 Yu, Simon C.H. II-353
 Yu, Tianli IV-602
 Yu, Yizhou III-533
 Yuan, Lu II-603
 Yuille, Alan L. III-195

 Zandifar, Ali III-482
 Zboinski, Rafal III-329
 Zelnik-Manor, Lihi II-434
 Zeng, Gang I-163
 Zha, Hongyuan IV-390
 Zhang, Benyu I-121
 Zhang, Hongjiang I-121
 Zhang, Ruofei III-355
 Zhang, Zhongfei (Mark) III-355
 Zhou, S. Kevin I-588
 Zhou, Xiang Sean I-336
 Zhu, Haijiang I-190
 Zisserman, Andrew I-69, I-228, I-242, I-390, II-85
 Zomet, Assaf IV-377